2006

Syntax-based Concept Extraction For Question Answering

Demetrios Glinos
University of Central Florida, glinos@cs.ucf.edu

SYNTAX-BASED CONCEPT EXTRACTION FOR QUESTION ANSWERING

by

DEMETRIOS GEORGE GLINOS
B.S. Trinity College, Connecticut, 1973
M.S. University of Central Florida, 1999
J.D. Georgetown University, 1976

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2006

Major Professor: Fernando Gomez

ABSTRACT

Question answering (QA) stands squarely along the path from document retrieval to text understanding. As an area of research interest, it serves as a proving ground where strategies for document processing, knowledge representation, question analysis, and answer extraction may be evaluated in real world information extraction contexts. The task is to go beyond the representation of text documents as "bags of words" or data blobs that can be scanned for keyword combinations and word collocations in the manner of internet search engines. Instead, the goal is to recognize and extract the semantic content of the text, and to organize it in a manner that supports reasoning about the concepts represented. The issue presented is how to obtain and query such a structure without either a predefined set of concepts or a predefined set of relationships among concepts.

This research investigates a means for acquiring from text documents both the underlying concepts and their interrelationships. Specifically, a syntax-based formalism for representing atomic propositions that are extracted from text documents is presented, together with a method for constructing a network of concept nodes for indexing such logical forms based on the discourse entities they contain. It is shown that meaningful questions can be decomposed into Boolean combinations of question patterns using the same formalism, with free variables representing the desired answers. It is further shown that this formalism can be used for robust question answering using the concept network and WordNet synonym, hypernym, hyponym, and antonym relationships.

This formalism was implemented in the Semantic Extractor (SEMEX) research tool and was tested against the factoid questions from the 2005 Text Retrieval Conference (TREC), which

operated upon the AQUAINT corpus of newswire documents. After adjusting for the limitations of the tool and the document set, correct answers were found for approximately fifty percent of the questions analyzed, which compares favorably with other question answering systems.

This work is dedicated to the memory of my father, George Demetrios Glinos, from whom I learned to love learning and to persevere in its pursuit, and to my wife Kathleen and our son George whose love and support, and particularly their good humor, sustained me throughout this experience.

ACKNOWLEDGMENTS

I wish to express my profound thanks and appreciation to my major advisor and friend, Dr. Fernando Gomez, for his patience and firm guidance in the direction of this research, and to Drs. Kien A. Hua, Carlos Segami, and Annie S. Wu, for their encouragement in this endeavor and for serving on the research committee.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ARDA	Advanced Research and Development Activity
IR	Information Retrieval
MUC	Message Understanding Conference
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
POS	Part of Speech
PT	Proposition Tuple
QA	Question Answering
SBCN	Syntax-based Concept Node
SEMEX	Semantic Extractor
TREC	Text Retrieval Conference

CHAPTER ONE: INTRODUCTION

This line of research involves the unsupervised extraction of knowledge from unrestricted text in a form that is suitable for question answering. More specifically, it involves the automated construction of knowledge structures, which we call "proposition tuples" and "syntax-based concept nodes", and the specification of the means for employing them together with the WordNet lexical database to support question answering from large text collections. This chapter presents the background and motivation for developing such knowledge structures. It also summarizes the principal claims set forth in this dissertation and gives an overview of its structure.

Central Claims

Open domain question answering (QA) from text involves the retrieval of answers rather than documents in response to questions posed of potentially large document collections. This sub-specialty of artificial intelligence research represents an important step towards natural language understanding and is quite different from information retrieval (IR), which is primarily concerned with retrieving entire documents from the collection in response to questions.

The principal goal of QA is to understand the information content of sentences such as this excerpt from a newswire article taken from the AQUAINT corpus:

> *Russia is testing a new nuclear submarine "Gepard" (Cheetah) with most advanced technologies on board for the Northern Fleet, which lost a nuclear-powered sub Kursk sinking in the Barents Sea during a naval exercise in August, Russian media reported Friday.*

sufficiently to be able to extract “in August” as the answer to the question, “When did the submarine sink?” Typically, this 41-word sentence appears in the context of a great number of other sentences that, taken together, constitute the textual information repository against which the question is posed.

To answer this question, one could take an approach based on keyword combinations and collocations, in a manner similar to that of a typical internet search engine. In this example, one could search for the occurrence of “submarine” and “sinking”, or their morphological roots or synonyms, within some number of words from each other, and if found, then examine a neighborhood of words around them for words or phrases that relate to time in order to find candidate answers for “when” questions. The advantage of such an approach is that the entire document, or each of its constituent paragraphs, sentences, or clauses, could be represented and stored as a simple “bag of words” or data blob that can be tokenized into words for input to the question answering algorithm.

If the above sentence comprised the entire text database, then one can see easily how the correct answer could be found using such an algorithm. However, it is not difficult to see that if the sentence structure were different, or if the sentence above were embedded in a document that contained a number of instances of “submarine” and “sinking” within close proximity of each other, and also a number of time words and phrases, that “Friday” or some other phrase could be returned by such an algorithm as the answer, or no answer at all, which would be incorrect.

The task, therefore, is to develop a method for question answering that is independent of sentence structure and word order. For this, one must go beyond the bag of words representation of text and instead to employ knowledge representation structures that reflect the invariant semantic content of the many ways to express the same proposition. Thus, for example, “Peter

saw a movie on Friday night," and "On Friday night, Peter saw a movie," represent the same underlying concepts and their relationships, so the knowledge structures representing these sentences should be the same. Similarly, one should be able to extract the same underlying structure from a sentence such as "Peter, who flew to New York from London, saw a movie on Friday night," although additional structures would be needed to express the non-defining relative clause that Peter flew from London to New York.

Thus, the task for QA becomes one of recognizing and extracting the semantic content of the input text, and organizing it in a manner that supports reasoning about the concepts represented. The critical issue, however, is what concepts and what relationships to encode in knowledge structures.

This dissertation reports on research into the construction and use of knowledge structures for question answering. We present two types of data structures for encoding conceptual information extracted from text, which we call "proposition tuples" (PTs) and "syntax-based concept nodes" (SBCNs), respectively, and argue that one can perform effective question answering using a network of such structures.

In particular, we present four main claims that are supported by this research.

The first claim is that we can let the input text documents themselves define both the concepts of interest and their interrelationships. We support this claim by presenting the syntax-based formalism for constructing PTs from the atomic propositions that are extracted from text, and by describing how PTs encode the n-ary relationships among the discourse entities that appear within the propositions. We further support this claim by describing the results of a test implementation in which such tuples were extracted and used successfully to answer questions.

The second claim is that the discourse entities within the PTs can be organized into a set of concept nodes that form a network of concepts based on "is-a" relationships among the discourse entities and concepts derived from them. We support this claim by describing how PTs may be organized into SBCNs, based on the discourse entities they cover, how additional concepts may be derived from concepts based on the discourse entities, how the SBCNs serve to index the PTs, and how the SBCNs are organized as a network of potentially disjoint sub-networks based on the many-to-many parent-child relationships among the concepts. This claim is further supported by the results of a test implementation, in which such networks were successfully constructed and used to answer questions.

The third claim is that questions can be expressed as Boolean combinations of PTs, with free variables representing the syntactical components for the desired answers. We support this claim by describing how questions may be decomposed into such combinations of tuples for various question types. This claim is further supported by the results of our test implementation, in which questions were successfully decomposed in the manner described, and then used successfully to answer questions.

The fourth claim is that the network of concept nodes can be used effectively to answer questions. We support this claim by describing how the network structure can be used to select the PTs of interest based on the discourse entities contained in the question, how the answer can be obtained by a pattern matching procedure operating on the question tuples and these PTs, how alternative question patterns can be derived, and how the synset, hypernym, hyponym, and antonym features of the WordNet lexical database (Fellbaum 1998) can be used to increase robustness in question answering. This claim is further supported by the results of our test

implementation, in which concept networks were constructed from text documents and questions were successfully answered.

Motivation

In this section, we present our motivation for this line of research. First, however, we present our understanding of the current research context. This serves as a backdrop for a discussion of recent work in developing QA systems and component technologies that have motivated our claims.

Current Research Realities

Consider the sentence, "HTIUE JVER SWQL TGS CFLP." Assuming that it is not encrypted, it is difficult to imagine either a human or a computer extracting any useful information from this sentence. However, to a computer, it is essentially no different from the sentence "Peter gave Mary the book," which has tokens containing the same numbers of characters as the original sentence, and which is readily understood by English language speakers.

What is the difference?

The human in this example has three key aids to understanding, which the otherwise unprepared computer does not. First, the human understands the tokens "Peter", "Mary", "gave", and "book". That is, the human has a "lexicon" containing the meaningful concepts of the language, in this case, English.

Second, the human in this example understands the grammar of the English language, so that he or she understands the sequence of tokens to mean that the entity represented by the concept "Peter" has transferred to the entity represented by the concept "Mary" a particular object of type "book". This is essentially a logical proposition relating the concepts that are being described, that is, the discourse entities.

And finally, the human is able to retain the proposition that Peter has given Mary the book in long-term memory, so that one can later answer the question, "What, if anything, did Mary receive from Peter?" For humans, this long-term memory is built up incrementally, so that if the hearer later learns that Peter has also given Mary an apple, then that information will be added to the concept for Mary and both the book and the apple will be "known" to have been received from Peter.

Now, if we could provide to a computer a complete lexicon of, say, English, containing all of the concepts that may be discussed in text and describing all of their interrelationships, and if we could further provide a complete grammar specification for the English language so that every well-formed sentence could be unambiguously interpreted, then question answering could be performed in a straightforward manner by parsing the question to determine the conceptual entities and relationships to search for, and then searching the lexicon of concepts for those entities and relationships.

The reality is, however, much different. While machine-readable dictionaries are now widely available, they do not contain the rich relationships needed for robust question answering. For example, such dictionaries do not ordinarily describe that a book is something that can be given posthumously as a devise by will, or that it may have a red leather-tooled cover, but that it cannot be breathed or poured. More importantly, as a practical matter a dictionary or lexicon

cannot possibly be complete, in the sense of containing every possible concept, as it will not contain entries for each person who is born, or each thought or concept that arises from a human's fertile imagination, any of which can nonetheless be the subject of discourse.

Although a complete lexicon is not available, useful lexical information is nevertheless available to researchers. One source for such information is the WordNet lexical database (Fellbaum 1998), which has become a primary research tool for English language computational linguistics. WordNet does not "define" words in the traditional manner of hardcopy dictionaries. Rather, it groups related words into "synsets" (synonym sets), where each synset represents a particular meaning or "sense" for the words contained therein. Each synset may have a "gloss" representing some useful information about the concept, although this is not generally adequate for a definition. Thus, for example, the word "give" has in WordNet one sense as a noun representing, according to its gloss, "the elasticity of something that can be stretched and returns to its original length." The word "give" also has 44 senses in WordNet as a verb, including its primary sense of "cause to have, in the abstract sense or physical sense", as well as senses for "sacrificing," for "performing," and many other distinct meaningful concepts. Thus, "give" and "sacrifice" are both contained in the same synset for the "sacrifice" concept, but "sacrifice" does not appear in the "performance" synset for "give". Overall, WordNet contains over 118,000 word forms for nouns, verbs, adjectives, and adverbs, and more than 90,000 different word senses, comprising over 166,000 form-sense pairs.

WordNet contains a number of pre-defined semantic relations that apply generally to the English language. These include: (1) synonymy, in which word senses are grouped into synsets; (2) antonymy, in which word senses are related by the symmetric relation of being opposites; (3) hyponymy and hypernymy, representing the "is-a" relationship both downwards and upwards;

(4) meronymy and holonymy, representing the "part-of" relationship both upwards and downwards; (5) troponymy, which is for verbs what hyponymy is for nouns, so that "madrigal" is a particular way to "sing"; and (6) entailment, describing necessary conditions between verb concepts, for example, "divorce" entails "marry", since one cannot divorce if one has not previously married. These relationships provide the basis for developing taxonomies of concepts, as some researchers have done (see Harabagiu et al. 2004), or for adding robustness to the search for answers to questions, as we discuss in Chapter 3.

The reality with respect to the language grammar is similarly less than ideal. The fact is that there is presently no complete specification of the English language that is sufficient for computational purposes. Many difficult problems remain, such as handling complex sentence structures, slang, metaphors, and prepositional attachment. Consider, for example, the sentence, "Peter saw the man on the hill with the telescope." Knowing nothing more than this sentence, one cannot determine who was on the hill and who had the telescope. In current research, such issues are addressed through statistical methods and discourse analysis. For example, if in the context of a paragraph describing Peter using his telescope, we encounter the sentence above, then it would be reasonable to attach the prepositional phrase "with the telescope" to the sentence subject noun phrase, namely, "Peter", but the opposite would hold if the topic discussed in the paragraph was that the man was an amateur astronomer setting up his equipment for a session. Semantic ambiguities also arise in other situations, for example, "Time flies like an arrow." The sentence is obviously well-formed, but without knowing more we cannot say whether the speaker is timing small insects or whether the speaker is describing the quick passage of time.

Although a complete grammar is unavailable, research in the field has coalesced around the use of part of speech taggers and partial parsers. Part of speech (POS) taggers assign to words of text a part of speech based on corpus-based statistical training and machine learning methods. Thus, for example, for the sentence "Peter gave Mary the book," the Brill tagger (Brill 1994) assigns the following tags "Peter/NNP gave/VBD Mary/NNP the/DT book/NN ./." indicating that "Peter" and "Mary" are proper nouns, "gave" is a past tense verb, "the" is a determiner, "book" is a singular noun, and the period is recognized as such. Current POS taggers, such as the Brill tagger, achieve approximately 95% accuracy (see Appelt and Israel 1999).

Partial parsers seek to achieve useful results by performing such parsing as can be reliably performed and at the same time by avoiding the difficult issues alluded to above. Thus, if a clause or prepositional phrase cannot be attached, then it is left "hanging", hence the partial nature of the parse. For the simple sentence in our example, the Cass partial parser (Abney 1996) takes POS-tagged input and produces the following simple parse tree:

```
<s>
 [c
  [c0
   [nx
    [person
     [fn Peter]]]
   [vx
    [vbd gave]]]
  [nx
   [person
    [fn Mary]]]]
 [nx
  [dt the]
  [nn book]]
 [per .]
</s>
```

The parse tree above shows how the POS-tagged tokens are grouped into noun and verb phrases (there are no prepositional phrases, relative clauses, or other syntactic constituents in this example), and how, even in this simple example, the noun phrase "the book" is not attached to the main clause, indicated by the "[c" marker and its corresponding "]" marker after "Mary".

POS taggers and partial parsers do make mistakes. Nevertheless, they are sufficiently reliable to serve as the basis for NLP research. For the research described in this dissertation, we employ both the Brill tagger and Abney's Cass partial parser as "black box" transformations that form early stages in the process of transforming text into logical forms.

Other researchers employ WordNet, POS taggers, partial parsers, and other tools in ways that differ from our own. Systems representative of the state of the art are described in the subsection that follows. Further subsections describe additional enabling technologies that have served as motivations for this line of research.

Question Answering Systems

Current QA systems primarily use keywords or other attributes extracted from the question both to restrict the subset of the document collection to consider and also to extract and answer from the documents and/or passages retrieved. (Voorhees 2005b) However, systems vary widely in their approaches for achieving these ends.

Thus, some systems, such as that described in (Habagiu 2004) are built around a named entity recognizer, which is used to classify the entities in a document, typically the noun phrases, according to a taxonomy of semantic categories, such as "person" and "location", that are built from WordNet noun and verb hierarchies. In this system, the semantic category of the expected

answer type is determined from the question, typically the head of one of the question phrases, and documents or passages are retrieved that contain the desired type.

Once the answer type is determined, keywords are extracted and passed to a document processing module that retrieves relevant passages. Candidate passages are ranked based on the number of occurrences of the expected answer type. The ranked passages are then passed to the named entity recognizer, CICERO LITE, to attempt to extract the answer based on the expected answer type. Answer extraction involves recognizing the answer phrase based on an expected answer pattern obtained from the question. For example, the question "*What is Iqra?*" is matched with the pattern <What is_*Question-Phrase*>, which corresponds to the answer pattern <*Question-Phrase*, "which means" *Answer-Phrase*>, from which the appropriate answer is obtained from the passage, *"Iqra , which means ' read ' in Arabic , was ...."*

If the answer is not found by CICERO LITE, the answer is sought in the answer taxonomy by abduction using a "theorem prover." Abduction is the ordered process of selecting the best explanation or hypothesis to explain the given facts or observed phenomena (see Abductive Inference Page 1997). As described, the theorem prover examines WordNet glosses for the concepts being considered and provides a mechanism for substituting one concept (term) for another. The concepts in the answer taxonomy were seeded using 100 concepts and populated using the multi-level bootstrapping technique reported in (Riloff and Jones 1999). The resultant taxonomy was manually verified. The authors report that the principal function of the theorem prover is to filter out incorrect answers. They also report that of the correct answers produced by their system, 71% were produced by named entity extraction, compared with 29% for the answer taxonomy.

The algorithms that we describe differ substantially from that described in (Harabagiu et al. 2004) in that they do not involve pre-defined taxonomies or a priori classifications. Both the named entity recognizer and the answer taxonomy described above involved such pre-defined classes, while the concept network that we describe is defined "on the fly" as the text is processed for the first and only time. Moreover, the authors also report that CICERO LITE was trained, indicating that its named entity recognition is statistically developed and corpus-based. By contrast, the algorithms that we describe are neither statistically based nor use any training corpus, and indeed, our SEMEX implementation of these algorithms does not incorporate such features.

The system described by (Yang et al. 2004), named QUALIFIER (Question Answering by LexIcal FabrIc and External Resources), contains modules for question analysis, named entity recognition, and anaphora and co-reference resolution. The question analysis module processes questions to determine the appropriate question class in a manually constructed ontology of 51 classes of 5 basic types: human, time, location, number, code, and object. Thereafter, the processing is different for definition questions than for factoid and list questions. Both lines of processing, however, are characterized by statistical methods and the use of external lexical resources.

For definition questions, the document retrieval subsystem returns documents containing the search target determined from question analysis. The documents are separated into sentences and anaphora resolution is applied to resolve appropriate pronoun references with the search target. Sentences containing the search target are retained. Word frequency statistics are obtained from the retained sentences as well as from snippets obtained from the Web using queries constructed from the retained sentences. The retained sentences are thereafter ranked

according to a statistical measure combining weights for the sentence derived from the input corpus and from the Web. To eliminate redundancy in definitions, a summarization technique employing MMR (Maximal Margin Relevance) is used to select non-redundant sentences from the ranked list. The authors report that they also employ sentence pattern heuristics to enhance the process by adjusting the weights for sentences that are heuristically likely to be "good" definitional sentences. For example, sentences containing "<*Target*>, a ….." or "<*Target*>, which is the …." are considered likely to be good definitional sentences.

For factoid questions, QUALIFIER models semantic content as QA "events" composed of different elements for time, location, object, action, etc. The QA event structure is constructed using the original query terms obtained from question analysis, obtaining a number of documents using a Web search engine, and then extracting the terms that are highly correlated to the original query terms. After WordNet is used to adjust weights and to introduce additional terms, the system computes lexical, co-occurrence, and distance correlations between terms to induce the event structure. The system uses "event mining" to generate useful association rules among the event elements. These rules are used to rank the relevant documents from the QA corpus. The answer is then extracted from the top-ranked passages using named entity recognition. Lexical features such as capitalization, punctuation, context, part-of-speech tagging and phrase chunking are used to perform named entity recognition. Anaphora resolution is performed using Charniak's parser to produce lists of all noun phrases in the text and all anaphors. Candidate associations are ranked and selected according to a binding algorithm that examines lexical features and assigns weights. The algorithm only considers candidate noun phrases occurring within three sentences preceding the sentence containing the anaphoric reference. List questions

are processed similarly to factoid questions, except that multiple answers are allowed and duplicate answers are detected and removed.

The algorithms that we describe differ from those described by (Yang et al. 2004) in that we do not employ statistical methods at all, nor do we utilize external lexical resources other than WordNet. Moreover, we do not restrict question analysis to a pre-defined ontology or classification hierarchy. Nevertheless, our systems are similar in that we propose to recognize discourse entities using many of the same lexical and semantic features that QUALIFIER employs for named entity recognition. However, even for discourse entity recognition, we do not propose to employ a predefined classification hierarchy.

The system described in (Litkowski 2003) describes an approach for producing a knowledge base that is queried to produce answers. The approach employs a system, DIMAP-QA, that consists of four major components: sentence splitting, parsing, discourse and sentence analysis, and question answering. Sentence splitting in this system is simply the detection of sentence boundaries. For parsing, the system employs a proprietary parser, which produces bracketed parse tree output with leaf nodes describing the part of speech and lexical entry for each sentence word.

The central component of the system is the discourse and sentence analysis module. Each sentence is treated as an event, and as each sentence is examined, data structures are built up to support discourse analysis: discourse markers (subordinating conjunctions, relative clause boundaries, discourse punctuation), verb lists to serve as the "bearers of the event for each discourse unit", data structures for anaphora resolution, and the event list. Noun phrases are recognized at this stage, and "semantic relation triples" are constructed. Such a triple consists of: (1) the discourse entity itself (the noun phrase), (2) a syntactic or semantic role relation, and (3) a

"governing word" with respect to which the entity stands in the semantic relation, which is usually the main verb of the sentence or the noun or verb to which a prepositional phrase attaches. Allowable roles include "SUBJ", "OBJ", "TIME", "NUM", "ADJMOD", and prepositions that head prepositional phrases. Thus, for the sentence, "Peter gave Mary the apple," the semantic relation triple for the noun phrase "Peter" would be <Peter, SUBJ, gave>. In this processing phase, anaphors are resolved as they are encountered and an attempt is made to assign a semantic type to the head noun, as distinguished from the semantic role that is used for constructing the semantic relation triple.

The output of the discourse and sentence analysis module consists of the semantic relation triples and four lists: (1) events (discourse segments), (2) discourse entities, (3) verbs, and (4) prepositions (representing semantic relations). Each discourse entity, verb, and preposition is then tagged with attribute information. For a discourse entity, its attributes include its segment; sentence position; syntactic role (subject, object, prepositional object); syntactic characteristics (number, gender, and person); type (anaphor, definite, or indefinite); semantic type (e.g., person, location, organization); coreferent, if any; whether it includes a number; antecedent; and a tag for the type of question it may answer (e.g., who, when, where, how many, and how much). For verbs, attributes include: segment, sentence position, sub-categorization type (from a set of 30 types); arguments; base form; grammatical role when used as an adjective. And for prepositions, attributes include: segment; semantic relation type; prepositional object; and attachment point. The author reports that such a database has proved cumbersome for tracing relations for certain classes of questions due to the flat structure of the database. Accordingly, the author presents a second approach in which document processing results in

marking up the input text document to produce an XML-tagged document whose nested tags encode relationships among the discourse entities.

Question answering using the document databases consists of detailed question analysis depending on the question type; coarse filtering of database records to select relevant sentences; extracting possible short answers from the sentences based on matching questions against database records; assigning ranking scores based on the "key elements" of the questions; and selecting the final answer based on the final rankings. For the XML-tagged version of the database, a question is analyzed and converted into an XPath expression; the XPath expression is used to query the XML file; and, if necessary, the nodes returned are scored and returned to the user. Forexample, a question such as *"What percent of Egypt's population lives in Cairo?"* becomes the XPath query, *"//segment[contains(.,'Cairo')]* //discent[contains(.,'percent') and @tag='howmany']", which should return all nodes containing the word "Cairo" that also contain descendant nodes that contain the word "percent", where the descendants have an attribute "tag" whose value is "howmany." The question may be answered from either database source. It is not necessary to answer the question from both sources. Either is sufficient.

Although the assignment of attributes to discourse entities in the DIMAP-QA system appears similar to what we describe in this dissertation, a closer examination reveals fundamental differences in the approaches adopted by the systems. The principal difference is that while the DIMAP-QA system creates its semantic relation triples as representations of logical forms, it allows only a limited number of predicates to be represented, namely, "SUBJ", "OBJ", "TIME", "NUM", "ADJMOD", and prepositions. Moreover, the semantic relation triples themselves have limited expressiveness, as they relate discourse entities to only a single word by such limited predicates. Further, the DIMAP-QA system contains data structures for verbs and prepositions

augmenting the data structures for the discourse entities and semantic relation triples, in order to capture the full range of semantic concepts and relationships that cannot be captured by discourse entity attributes alone. Finally, although details are not given in the article, it appears that question answering involves keyword and/or pattern matching between question and candidate answer without augmentation using other lexical resources.

By contrast, the logical forms that we describe are not limited in the types of predicates that are permitted. Basically, any verb can serve as a predicate. As a result, we claim to be able to construct a meaningful concept network composed solely of discourse entity nodes containing, among other features, representations of the logical forms within them. Further, the logical forms that we propose ("proposition tuples" or "PTs") explicitly relate the various sentence constituents (subjects, verbs, objects, and modifiers) to one another. And finally, the question answering mechanism that we propose employs WordNet to expand the query space beyond the words in the question to reach antonyms, hypernyms, and other related concepts on the path towards answer discovery. Overall, we believe that the DIMAP-QA system reflects a design approach that was centered on verbs as the focus of discourse analysis, whereas the system that we propose is designed around the concepts that are to be related in the concept network.

The spectrum of system approaches may be observed from the systems described below which, while distinguishable from the approach or our own research, serve to illustrate the considerations that researchers in this field incorporate into system and algorithm design.

In (Greenwood et al. 2003), the authors describe a system in which the input text is processed to produce a set of "quasi-logical forms" (QLFs), from which a semantic net or discourse model is created incrementally for all the entities and relationships that are represented in the QLFs. With the addition of a special semantic predicate to represent the question variable,

questions are also processed into the same QLF form. For example, the question *"Who wrote Hamlet?"* produces the QLF: *"qvar(e1), qattr(e1,name), person(e1), lsubj(e2,e1), write(e2), time(e2, past), aspect(e2,simple), voice(e2,active), lobj(e2,e3), name(e3,'Hamlet')"* For this system, the QLF for the question is added to the discourse model for the text and the answer is obtained by resolving the reference for the question variable.

The authors of (Gaizauskas et al. 2005) present what they term "a shallow multi-strategy approach" for question answering, consisting of a cascade of three strategies in which, if an answer is found at any stage, then succeeding stages are not executed. The first stage consists of matching surface text patterns learned from the Web using the methods described in (Greenwood and Saggion 2004). The patterns are learned in a two-step process: first, performing a web search based on sets of known answers for question forms such as *"When was X born?"* and extracting from the documents retrieved the declarative patterns in which the answers occur; and second, filtering out patterns of little value by performing a subsequent web search using the retrieved patterns and a second, independent set of answers for the same question type. If the first stage does not produce an answer, then semantic type extraction is attempted. In this second stage, the expected answer type is determined from a predefined taxonomy, the information retrieval engine retrieves a set of passages for this answer type, and then all entities of the expected answer type are extracted from the documents. The answer is selected to be the candidate answer with the greatest overlap with the question. The final stage, which is executed only if neither preceding stage produces an answer, represents a catch-all for question types not covered by the preceding stages. At this stage, the document set is searched for occurrences of hyponyms of a key word extracted from the question. For the question *"What grapes are used in making*

wine?" the hyponyms of "grape", such as "Concord grape", are obtained from WordNet and are used to obtain matches against the document set.

In (Ahn et al. 2005), a semantic formalism is presented in which Discourse Representation Structures (DRSs), containing ordered pairs of discourse entities and conditions, are augmented with answer literals and are applied to both the question and the input text. Answer extraction is obtained through Prolog unification, where discourse entities in questions are represented by Prolog variables and in the text passages by Prolog atoms. The authors also present an alternative approach in which a web query is constructed from the question phrases and additional keywords based on the question type. Sentences in the returned text snippets are processed in pairs through a Longest Common Substring (LCS) dynamic programming matrix to produce a set of relevant nodes. All possible pairs of nodes are examined and labeled with the "EQUIVALENT", "OCCURS_IN", or "HAS_OCCURRENCE" relation that applies. For example, the node *{Songiform, Encephalopathy}* OCCURS_IN *{Transmissible, Spongiform, Encephalopathies}*, which encodes the entailment relationship of the latter concept by the former.

A system in which different strategies were applied to different types of questions is described in (Schone et al. 2005). For definition and "how many" types of questions, a "cascade of filters approach" (CFA) was employed. These filters included: (i) measures of distance between question keywords and target numeric quantities, if any; (ii) the application of simple template match filters derived from the question; (iii) semantic rules that exclude candidate answers based on sentence component mismatches; and (iv) sentence and question trigram matching. The answers were the values that survived the filtration process. For other kinds of questions, a "knowledge-graph induction" strategy was employed. This strategy involved a deep

syntactic parse, followed by entity identification and classification. A graph builder then constructed an indexed, directed, attributed entity-relationship graph using noun phrases as entities, verb phrases as relationships, and quantifier, prepositional phrases, and adjectives into attributes. Thus, for example, for the sentence *"Johan Vaaler invented the paper clip 90 years ago,"* there would be a node "johan_vaaler" with directed arcs to entities "johan" and "valer", and a relationship "invented" connecting it to the node "paper_clip". Questions were processed similarly to produce a set of objects of interest, which were used to identify candidate answers from the nodes in the graph. The graphs rooted at the candidate entities were then tested for the reachability of the remaining objects of interest through appropriate directed arcs. An answer was deemed found if all needed components could be reached through this graph search, or if none, then the algorithm returned the candidate with missing components that were closest by a distance measure. The authors also presented a web-based validation strategy seeking to confirm that question words occur in close proximity to proposed answers.

The systems described above demonstrate widely varying approaches to the question answering problem. The various systems represent different combinations of relative complexity in document processing, question analysis, and answer extraction. Each has its own strengths and weaknesses, as more fully described by its authors. In the chapters that follow, we present a formalism for question answering that we feel represents a good balance of complexity in these areas, and which we believe supports a simple, but elegant, method for question answering and a growth path for reasoning beyond simple question answering.

Information Extraction

Techniques and methods employed in question answering have been based in part on previous and ongoing work in information extraction in general and pattern matching in particular. One important influence on QA has been the work of Ellen Riloff, which has evolved from previous template-matching efforts in support of Message Understanding Conferences (MUCs, the predecessors to TRECs), and which has evolved over a number of iterations. This body of work is included here because the pattern matching mechanism described has become a staple in the IE and QA fields, and indeed, pattern matching is employed in the question answering algorithms that we present in this dissertation.

In (Riloff 1993), the author presents AutoSlog, a system for automatically generating a "concept node" dictionary for use by the UMass system that performed among the best in the MUC-3 competition. For MUC-3, the UMass system used a dictionary that was entirely crafted by hand, requiring approximately 1,500 person-hours to create. For MUC-4, the performance of this handcrafted dictionary (augmented by 76 nodes generated by AutoSlog) was compared against that of a dictionary that was generated entirely by AutoSlog, which required only 5 person-hours to create.

The knowledge extracted by the system consisted of instantiated concept nodes, which are essentially case frame templates that are triggered by specific keywords (called "conceptual anchor points" or "trigger words"), and which are activated by the specific linguistic context of the trigger words and which must satisfy specified constraints. Concept nodes have slots for the information that is extracted when all constraints are satisfied and the node is instantiated. Thus, for the terrorism domain used for MUC-4, a "kidnap" concept node would have slots for the

victim, the perpetrator, the location, and so on. While concept nodes in general may have any number of slots, the nodes generated by AutoSlog had only one slot, so that a number of different concept nodes would be instantiated. These separate concept nodes were presumably subsequently pieced together by the system to produce the instantiated answer template for the target text, although this aspect of the system was not discussed in the article.

AutoSlog employed a set of thirteen heuristics for generating proposed concept nodes. These heuristics were in the form of linguistic patterns or templates that were instantiated when triggered by training example text-answer pairs. Thus, for example, for the text "the diplomat was kidnapped" and the answer "the diplomat", AutoSlog would instantiate the "<subject> passive-verb" template to return a concept node in which "kidnapped" in the passive context is the conceptual anchor point. The heuristic set included four patterns in which the extracted information was the subject, six in which it was the direct object, and three in which it was the object of a preposition.

In (Riloff and Jones 1999), the authors present a multi-level bootstrapping method for constructing both the set of domain-specific extraction patterns and a semantic lexicon, using only a set of seed words for the domain and an unannotated training corpus. Significantly, the corpus is not even pre-classified with respect to domain relevancy, as was required for previous versions. The inner bootstrapping loop proceeds as follows. First, AutoSlog is used exhaustively to generate the complete set of extraction patterns that will extract every noun phrase in the training corpus. Since this generally results in a very large set of extraction patterns, all of the extraction patterns are scored using an "RlogF" metric, which rewards both patterns that extract terms in the lexicon with high frequency and patterns that correlate highly with the terms in the lexicon. Next, all of the noun phrases that are extracted by the highest

scoring pattern are added to the lexicon, and then the re-scoring, selection, and lexicon augmentation are repeated until some threshold conditions or loop count are met. As bootstrapping proceeds, the set of selected extraction patterns and semantic lexicon both grow.

This method has the side effect that a few bad entries quickly infect the dictionary and skew the extraction pattern scoring by rewarding extraction patterns that extract erroneous terms in the lexicon. To prevent this phenomenon, an outer or meta-bootstrapping mechanism is included. At this level, once the inner bootstrap process terminates for one cycle, only the top five lexical entry additions are retained and added to the permanent semantic lexicon, and the inner bootstrapping process is started afresh with this augmented lexicon. The lexical entries are scored according to a function that rewards those entries that are extracted by the greatest number of extraction patterns, modified slightly by the scores for such patterns. After the final meta-bootstrapping run, the extraction patterns produced by the last iteration and the permanent semantic lexicon at that stage are selected as the dictionaries for information extraction in the domain. The resulting lexicon does need to be manually reviewed by a human to weed out bad entries, but it is argued that this can be done quickly.

Tests were conducted using this multi-level bootstrapping algorithm where the meta-bootstrapping loop was run for 50 iterations and the inner loop was run until 10 new patterns were selected, more or less, depending on some threshold criteria that were designed to generate a variable number of extraction patterns based on their reliability. Results against the MUC-4 terrorism database and a set of corporate web pages showed good recall and precision, and indicated, according to the authors, the viability of this approach.

The system described in (Riloff and Jones 1999) was employed in (Harabagiu et al. 2004) to generate the answer taxonomy used in that system.

Extraction patterns of the type used in Auto-Slog were also used in (Hildebrandt et al. 2004), which reported on a system that participated in TREC 2003. Although the system operated on all the different question types (factoid, list, and definition), only the processing of definition questions was discussed in the article. The authors presented a three-part strategy for answering definition questions. For the first mechanism, the authors developed for the limited *definitions* domain a set of eleven extraction patterns ("surface patterns") that, when instantiated with the target term "*t*", retrieved noun phrases that corresponded to the free variable "*n*". The set of patterns employed by the authors is shown below:

NP_1 be NP_2	where $NP_1 = t$, $NP_2 = n$
NP_1 become NP_2	where $NP_1 = t$, $NP_2 = n$
NP_1 *v* NP_2	where $NP_1 = t$, $NP_2 = n$, $v \in$ biography-verb
NP_1 , NP_2	where $NP_1 = t \vee n$, $NP_2 = t \vee n$
NP_1 NP_2	where head(NP_1) $\in$ occupation, $NP_1 = n$, $NP_2 = t$
NP_1 (NP_2)	where $NP_1 = t$, $NP_2 = n$
NP_1 , (also) known as NP_2	where $NP_1 = t \vee n$, $NP_2 = t \vee n$
NP_1 , (also) called NP_2	where $NP_1 = n$, $NP_2 = t$
NP_1 , or NP_2	where $NP_1 = t$, $NP_2 = n$
NP_1 (such as | like) NP_2	where $NP_1 = n$, $NP_2 = t$
NP (which | that) VP	where NP = *t*, VP = *n*

The authors "precompiled" a list of all the informational nuggets (the noun and verb phrases corresponding to "*n"* in the patterns above) corresponding to every entity mentioned in the AQUAINT data set. In this manner, the authors "automatically constructed an immense relational database containing nuggets distilled from every article in the corpus." This permitted them to answer definition questions by database lookup of the target term extracted by a simple pattern-based parser from the question.

For their second line of attack, the authors extracted noun phrases (NPs) from the Miriam-Webster online dictionary definition for the target term that was extracted from the question. These NPs were then used with an IR engine to retrieve the top 100 documents in the

data set containing those NPs. The documents were split into separate sentences and sentences that did not contain the target term were discarded. The remaining sentences were scored and ranked, and the answers were taken to be 100-character windows around the target term. Where answers were returned by both the first and second methods, an answer merging module was employed to detect and remove redundancies in the returned responses.

The third line of attack, which was used only if no answers were found by the preceding two methods, was to do a traditional document retrieval based on the target term extracted from the question. All document sentences containing the target term were shortened if necessary and returned as the answer.

This article is a good example of the strong influence traditional document retrieval and information extraction technologies continue to exert over current question answering systems. We argue for taking the next step, from indexing likely answer components to actually constructing a network of concept nodes containing the answer information. The key to such a network is in the knowledge representation scheme for the network. This is discussed in detail in the next section and in Chapter 2.

Knowledge Representation

In order to reason about concepts, and in particular to answer questions about the objects of discourse, one must have some data structure representing the concepts being examined and some means for representing the semantic relationships among concepts.

In (Gomez 2000), the author offers candidate constructs for representing concepts and the relations among them. The constructs are built from logical forms consisting of predicates and

their associated thematic roles. Thematic roles in this context represent not only standard semantic roles, such as ACTOR for the subject that takes the action described by the predicate, or THEME for the thing affected by the event, but also temporal and locative adjuncts. Thus, for the sentence, "Robins eat berries," the ACTOR is "Robins", the theme is "berries", and the action is, of course, "eat."

For each such logical form, an "action structure" (or "a-structure") is constructed for the predicate, containing a slot for each argument and the specification of the thematic role played by each such argument. Every argument is indexed as a node containing a reference to the a-structure that spawned it, and which is linked through "is-a" relationships to the other objects in the hierarchy of concepts. In the example above, concept nodes would be created for both "Robins" and "berries." This knowledge representation system is sufficiently rich to be able to represent "phrasal concepts", which the author argues is necessary for representing the objects of everyday discourse. For example, consider the sentence, "Spiders that live under water breathe from air bubbles," for which the author argues that "spiders that live under water" needs to be represented as an atomic object in memory so that subsequent information about it can be integrated on that node. In other words, the phrasal concept must be preserved as a single entity, so that its relations to other objects may be represented.

The author further argues that the knowledge contained in such a representational scheme can be acquired automatically and discusses how inferencing may be performed using such structures. A method in which a corpus is used to provide feedback for the interactive development of predicates is described in (Gomez 2004a). And in (Gomez 2004b), the author describes how the WordNet noun ontology can be reorganized and modified to support better its use in determining whether a particular thematic role is instantiated for a given verb predicate.

The key to the overall scheme, however, is the requirement for a semantic interpreter that assigns thematic roles to parsed sentences.

For our own research, we have adopted the notion of maintaining knowledge representation structures for the logical form, as indeed did (Litkowski 2002). In our case, however, while the predicates correspond to the verbs encountered in the text being examined (as they did for (Gomez 2000)), the arguments of the predicate do not correspond to the thematic roles required by the verbs (predicates). Instead, they consist of the syntactic roles played by the discourse entities entailed in the proposition. In this manner, we relax the requirements on semantic interpretation so that less than the full rigor needed to assign thematic roles is required. As a consequence, our system can operate at the syntactic level, using part of speech tagging, partial parsing, and our own post-processing without requiring any pre-defined lexicon or hand-crafting of action structures for particular predicates. Our approach also differs in that the knowledge structures representing the discourse entities are not only related to one another through "is-a" relationships, but also through the predicates encapsulated in the proposition tuples in which they appear. They are also related to each other and additional concepts through links to WordNet, whose structure encodes additional relations, such as synonymy and antonymy, described previously.

Other structures have also been proposed for encapsulating textual knowledge. In (Montes-y-Gomez et al. 2000), for example, the authors describe the use of conceptual graphs to represent document content for information retrieval and text mining applications. In the study reported, the authors constructed conceptual graphs for the titles of a database of 512 scientific articles and compared the graphs for similarity. As described, a conceptual graph consists of a network of two types of nodes: concept nodes and relation nodes, where concept nodes represent

entities, attributes, or events (actions), and relation nodes represent the (binary) relationships between pairs of concept nodes. A semantic analyzer is used to construct a conceptual graph from the tagged and syntactically parsed input text. Thus, the sentence "John loves Mary," is represented as the conceptual graph "[John] — (subj) — [love] ® (obj) ® [Mary]", where the concept nodes are in brackets and the relation nodes are parenthesized. In this formulation, the relations must be predefined, and indeed the authors report that only a limited set was employed, although there were plans to include domain-specific semantic relations.

This contrasts with the approach taken in our own line of research, in which we avoid the requirement to predefine the available relations by letting the input text itself define the relations from the predicates of the atomic propositions that are derived from them. Moreover, the concept network that we describe consists of only one type of node, based on a discourse entity. And further, the proposition tuples that we propose encode the n-ary relations among the syntactical components of such propositions, not just binary relations.

CHAPTER TWO: CONCEPT EXTRACTION

Our approach for concept extraction from text was guided by a number of motivating constraints. Chief among them was that the formalism should be based on features of the document set that may be computed easily on a single pass. A closely aligned motivation was that we should let the document set itself establish the relations among the entities of interest. We also desired an approach that would be robust with respect to word choice.

In this chapter, we present our proposed knowledge representation structures, the "proposition tuple" (PT) and the "syntax-based concept node" (SBCN). In the sections that follow, we present first our motivation for using syntax rather than semantics as the basis for our concept representations. Then, in succeeding section, we define the PT to represent the logical forms obtained from input text and the SBCN as a structure to organize and index a set of PTs to represent knowledge about a single concept or phrasal concept. Finally, we discuss how the set of SBCNs operates as a network of concepts and how we extend its power by linking SBCNs to WordNet noun hierarchy entries.

Why Syntax-Based?

Ideally, we would like to base our knowledge structures on the direct representation of the semantic content of text. An example from (Allen 1995) is this collection of sentences:

> John broke the window with the hammer.
> The hammer broke the window.
> The window broke.

In each of these cases, *John, the hammer,* and *the window* play the same semantic roles, namely: *John* is the actor, *the window* is the object of the action (breaking), and *the hammer* is the instrument used in the action.

As more fully described in (Allen 1995), semantic roles may include: CAUSAL-AGENT (the object that caused the event), INSTRUMENT (the force or tool used in causing the event), THEME (the thing affected by the event), EXPERIENCER (the person involved in perception or a physical/psychological state), BENEFICIARY (the person for whom an act is done), AT-LOC (the current location), AT-POSS (the current possessor), TO-LOC (the final location), PATH (the path over which something travels), CO-AGENT, and CO-THEME, among others. Thus, for the sentence, "The ball rolled down the hill to the water," the roles are: THEME (the ball), PATH (down the hill), and TO-LOC (to the water), where the action is one of "rolling" (Allen 1995). Similarly, for the sentence, "Jack is tall", the action is "is tall" and Jack is the THEME, but there is neither PATH nor TO-LOC nor any other role present in this sentence (Allen 1995).

From these simple examples, it is evident that the set of permissible roles and their syntactic realizations is different for each class of verbs. Moreover, one must have a lexicon or a comprehensive ontology to instantiate a structure based on such semantic roles. Thus, in (Gomez 2001), where it is argued that the over 11,500 WordNet verb classes (synsets) (see Fellbaum 1998) can be coalesced into a much smaller number of abstract semantic predicates, the differentiators between an abstract predicate and its included subpredicates depend on the existence of such lexical knowledge. For example, for the abstract semantic predicate *change-of-location-by-animate*, the instrument of the subpredicate *drive-a-vehicle* is always a vehicle, but for the generic abstract predicate it can be an animate, an animate body part, or something similar (Gomez 2001). Thus, to know whether or not the candidate *instrument* "Chevy" is a

"vehicle" a lexicon or ontology is needed. Indeed, (Gomez 2001) describes how it is necessary to reorganize WordNet verb classes and extend WordNet's noun ontology to support abstract predicates, as well as to define individual predicates for each abstract predicate.

It is argued here that these actions present a daunting task made necessary by the deep knowledge required for semantic-based knowledge representation. The fact that such actions are required indicates strongly that the state of the art in tools and lexical resources is not presently sufficiently mature to support the deep semantic-based interpretation of text.

We avoid these difficulties by taking a different approach.

The approach we have adopted is to infer semantics from syntax. The relationship between semantics and syntax is well known. Indeed, (Gomez 2001) cites several authorities to support the view that the syntax of many verbs is determined by their meaning. Although we recognize that the relationship between semantics and syntax is not simple, we approach the problem as one of determining the structure of what is inside a "black box" by studying its outputs. This is an accepted practice in engineering, one that we argue may be used to advantage in the present context.

Consider the sentence, "Peter threw the ball" and suppose that we are given that "Peter" is the subject, "threw" is the verb, and "the ball" is the direct object. What, then, do we know? Without any lexicon or ontology to define any of the terms for us, we "know" the following things: (1) "threw" is some kind of action; (2) "Peter" is someone or something capable of throwing; and (3) "the ball" is something that can be thrown. Moreover, if we can store this knowledge in some appropriate structure, we can answer questions such as "Who or what threw?", "Who or what threw the ball?", "What was thrown?", and "What did Peter throw?".

We believe that this is quite a store of knowledge garnered from just four words, none of which is defined.

What made this possible, in our view, is that: (1) we were provided with the *syntactic* roles (subject, verb, and direct object in this example) for the lexical objects, and (2) the relationships among these objects made sense to us because we assumed that the input text was a well-formed sentence. Given these two things, we take the relationships among the lexical objects as they are presented in the sentence, and infer from them corresponding relationships among the concepts in the "black box" that are represented by those lexical objects. Further, we believe that the well-formed nature of input sentences is a safe assumption for most text of interest, particularly in this day and age of grammar and spell-checkers, although we recognize that there is much written and spoken text that is "ungrammatical" and that grammar and spell-checkers have their own faults and limitations.

What makes this approach interesting, in our view, is two things. First, we believe that the current state of the art in tools and resources supports the reliable determination of syntactic roles for much unrestricted text. And second, we believe that the task of organizing syntactic role assignments into knowledge structures is a task that can be fully automated without requiring either a predefined lexicon or a training corpus. In the remaining sections of this chapter, we describe our proposed knowledge structures in detail and how they may be organized into a network. In Chapter 3, we describe how the nodes and the network may be used for question answering. And in Chapter 4, we report test results from an implementation of these concepts in a tool that we have developed to test our theories.

Proposition Tuples

With the preceding as a guide, we propose the notion of a proposition tuple (PT) as the basic unit of knowledge that is mined for question answering.

The PT is designed to accommodate the necessary syntactical components of the basic clause (sentence) patterns for the English language. As described by (Kries 2004), there are seven basic clause patterns, which are defined by their constituents among (coincidentally) seven components: (1) subject ("S"); (2) verb ("V"); (3) indirect object ("IO"); (4) direct object ("DO"); (5) subject complement ("SC"); (6) object complement ("OC"); and (7) adverbial complement ("AC"). This classification scheme ignores adverbials ("A"), which are considered "optional" in that they could easily be eliminated from a sentence and the resultant sentence would still make sense. Taking an example from (Allen 1995), we consider the sentence, "Jack ate the pizza by the door." If we eliminate the prepositional phrase "by the door", which serves as an adverbial, we are left with "Jack ate the pizza," which is just a more general assertion about the same situation. The same prepositional phrase, however, serves as an adverbial complement in the sentence, "Jack put the box by the door," since the sentence does not make semantic sense without it. For our purposes, we treat all adverbials as if they are adverbial complements, since we operate without a lexicon and therefore we do not have the basis for making the semantic judgment necessary to distinguish the two situations.

Based on this list of components, then, we define the PT as a logical form consisting of an atomic predicate and its arguments. For a simple sentence, the main verb may be taken as the predicate, with the remaining syntactical components as its arguments. Thus, for *"Peter threw the ball,"* we have the predicate *threw(Peter,ball)*. This is not generally the case for more

complex sentences. Consider, for example, the following sentence from the AQUAINT corpus used in TREC: *"Tourism is one of the major industries in Port Arthur, a town at the southern tip of the island."* The implementation described in Chapter 4 successfully splits off the apposition, producing two simpler sentences: *"Port Arthur is a town at the southern tip of the island,"* and *"Tourism is one of the major industries in Port Arthur."* In this form, each of these simpler sentences can now be expressed as a logical proposition from which the predicate and its arguments may be easily extracted. Similar splitting may be performed for sentences containing subordinate clauses, non-defining relative clauses, coordinations, and similar syntactical constructs.

Given that the input text can be decomposed into distinct propositions through sentence splitting as defined above, we define the proposition tuple for a given proposition as the 6-tuple

<subject,verb,gerinf,modifiers,indirect,direct>

where "subject" refers to the noun phrase representing the subject of the sentence, "verb" refers to the main verb phrase, "gerinf" represents any gerund or infinitive form, "modifiers" refers to adverbials and adverbial complements, typically prepositional phrases, "indirect" refers to the indirect object, if any, and "direct" refers to the direct object, if any.

To facilitate question answering, our PT definition varies in minor respects from Kries's list of syntactical roles. In particular, we separate gerunds and infinitives from the main verb component so that we can differentiate the propositions extracted from, for example, "Peter likes Mary," and "Peter likes to eat fish." While both "Mary" and "to eat fish" are appropriate answers to the question, "Who or what does Peter like?", only the latter is an answer to the question, "What does Peter like to eat?" We also choose to encode subject complements (e.g., "The mind is complex.") as modifiers, since their processing for pattern matching is the same, but we

encode object complements (e.g., "We consider her the best.") as separable components of the direct object.

We observe also that proposition tuples capture the n-ary relationships among all of the non-null syntactical components of the input proposition. By capturing all of the syntactical elements in the same logical form, the PT thereby encodes the binary relationships between subject and verb, object and verb, subject and object, and between any of the modifiers and other sentence components or even other modifiers. It also encodes higher order relationships, such as a tertiaary relationship between a given subject, object, and verb, and so on. As an encoding mechanism, the PT thus represents an efficient means of representing the set of all such relationships.

As an example, consider the following sentence, which is also taken from the AQUAINT corpus: *"As a professor at the University of Chicago in the early 1940s, Fermi designed and built the first nuclear reactor that later was put into use for research into nuclear weapons."* There are two propositions contained in this sentence, so that there are, correspondingly, two proposition tuples, the first of which is:

Proposition tuple #1:

subject:	Fermi,
verb:	designed
gerinf:	null
modifiers:	as a professor; at the University of Chicago; in the early 1940s
indirect:	null
direct:	the first nuclear reactor that later was put into use for research into nuclear weapons

where the second proposition tuple is identical to the first, except that the verb is "built". The three verb modifiers in this example, all of which are prepositional phrases, are separated by semicolons in the modifier field, indicating that they are separate and distinct objects. This

supports question answering using any combination of these modifiers, from none to all, in any order. Also noteworthy is the capture of an entire phrasal concept as the direct object. Phrasal concepts form the basis of the concept network discussed in the next subsection.

We also note that the logical form represented by a PT differs from the logical form that others may extract from the same sentence. Consider again the sentence "Jack is tall", from which (Allen 1995) extracts the logical form *is-tall(Jack)*. The corresponding PT for this sentence is <*Jack,is,__,tall,__,__*>, which corresponds to the logical form *is(Jack,__,,tall,__,__)*. The important difference here is that for the PT the subject complement ("tall") is retained as a separate lexical object and does not become part of the logical predicate. We do this for two reasons. First, where the subject complement is a noun phrase, as in "Elizabeth is Queen of England", we can take advantage of the symmetric qualities of the definitional "is" predicate to construct *is(Queen-of-England,__,Elizabeth,__,__)* from *is(Elizabeth,__,Queen-of-England,__,__)*, and *vice versa*, where needed during question answering. Second, the verb "is" (a form of "to be"), is a copular verb, indicates that this is a definitional proposition. The copular verbs include "seems", "appears", "looks", "sounds" and other verbs of perception, one of which must be present to have a subject complement (see Kies 2004).

Syntax-Based Concept Nodes

When we consider what is involved in answering a question, the first thing that comes to mind is "What is the question about?" In other words, we wish to know the question's subject or *target*. The target is most often a noun phrase describing a person, place, or thing, as in "Who is Alberto Tomba?" But it can also be an abstract object, as in "What does 'consanguinity' mean?"

However, whether tangible or intangible, abstract or concrete, the target is always a *concept*. Thus, “Peter”, “birds that live in the Antarctic”, and “trigonometry” are all concepts that can be the subject of discourse, and therefore, the targets of questions.

Examining questions such as these, we observe that they generally ask us to describe the characteristics of concepts and the relationships of concepts to one another. Accordingly, it would be useful to have a knowledge structure that contains all of the information that we possess concerning a single concept. If we had such a structure, we would be able to examine that structure to answer any question about the concept it represents.

The question arises, therefore, what should such a structure contain?

Given that a proposition tuple represents the interrelationships among the non-null syntactical components of the associated proposition, the tuple is relevant for answering questions that pertain to any of the discourse entities contained in any of these components. Accordingly, as an indexing mechanism, we wish to create “concept nodes” for all such entities and associate the tuple with each such node. For these purposes, a discourse entity is taken to be each noun phrase contained in a subject, indirect object, and direct object, and the noun phrase objects of prepositional modifiers. Direct quotes, subordinate clauses, and other phrasal concepts are not further decomposed at this stage. For example, given the sentence *“Peter enjoyed reading the book that Mary recommended,”* we generate the simple proposition tuple *<'Peter','enjoyed','reading',__,__,,'the book that Mary recommended'>*, from which we create concept nodes for 'Peter', 'Mary', and 'the book that Mary recommended'.

We also wish to construct the concept set to support complex question answering. Consider, for example, a document containing the sentences:

The Russian submarine Kursk sank in the Barents Sea.

The Russian submarine Kursk sank in deep water.

If the question were *"Where did the submarine sink?"* both *"in the Barents Sea"* and *"in deep water"* could arguably be acceptable answers. However, if the question were *"In what sea did the submarine sink?"* then only the first would be acceptable. Since the propositions represented by these sentences are syntactically identical, we distinguish them by postulating a mechanism by which we recognize *"the Barents Sea"* as an instance of the *"sea"* category contained in the question.

The method for distinguishing these cases is to encode all "is-a" relationships as explicit parent-child relationships among the corresponding concept nodes. Thus, if the document also contained a sentence stating, essentially, that the Barents Sea is a sea, this would be sufficient. However, we observe anecdotally that neither ordinary conversation nor newswire text typically contains such explicit categorizations. Therefore, as a corollary to the rule above, we postulate the need for deriving parent-child node relationships from individual concept nodes so that, in the example above, *"the Barents Sea"* is a *"sea"* because the lowercase head noun *"sea"* can be found to be a common noun, of which *"the Barents Sea"* is therefore an instance.

This feature also permits us to accommodate "phrasal concepts", which occur commonly as objects of discourse whose relationships to other concepts is essential to understanding (see Gomez 2000). For example, "birds that live in the Antarctic" is a phrasal concept for the reason that it stands for a single, atomic discourse object in the sentence, "Birds that live in the Antarctic are endangered by factory fishing." We can discuss, for example, how such birds survive in Antarctic conditions, as well as their prospects for survival. Moreover, we can establish the "is-a" relationship upwards to the concept for "birds" by analyzing the syntactic form of the phrase to find the head noun ("birds") and the defining relative clause. But the

propositions that involve this phrasal concept also say something about birds in general,, for example, that one kind of bird is a bird that lives in the Antarctic. Thus, a structure that captures the "is-a" relationship between these two concepts permits one to reason about the relationship between them.

Table 1 list a number of parent-child derivations that we have identified and which the system described in Chapter 4 has implemented. The arrows in the "Example" column of the table run from the child concept to the parent concept. These derivations are repeatedly applied to a concept until it cannot be decomposed further. Thus, for the concept "space shuttle Discovery", application of the last derivation followed by the one above it, produces the links that establish that "space shuttle Discovery " is-a "space shuttle" is-a "shuttle", which has the beneficial effect of rooting a set of concepts in a common noun form which, in this case, can be found in WordNet. This permits use of word synonyms when answering queries concerning the nodes so rooted, as more fully described in Chapter 3. No doubt additional derivations may be found; however, the set presented serves to illustrate the mechanism.

Table 1. Parent-Child Concept Derivations

Derivation	Example
Common noun-proper noun	"space shuttle Atlantis"-->"space shuttle"
Common noun-common noun	"oil tanker" --> "tanker"
Proper noun with preposition	"King of England" --> "king"
Common noun parent of proper	"Nobel Prize" --> "prize"
Proper noun-common noun	"Baldwin piano" --> "Baldwin", "piano"
Adjective-common noun	"Russian submarine" --> "submarine"
Multiple proper names	"Peter and Paul" --> "Peter", "Paul"
NP coordination	"fish and chips" --> "fish", "chips"
NP following preposition	"jar of beans" --> "jar"
NP following adjective/adverb	"fast cars" --> "cars"
NP following prep in proper noun	"Nobel Prize for Physics"-->"Nobel Prize"

Possessive form	"Mary's car" --> "car"
Business entity suffix	"IBM Corp." --> "IBM"
Comma-separated location	"Normandy, France" --> "France"
Concept prefixed by ordinal	"49th pageant" --> "pageant"
Title before proper name	"Miss Lara Dutta" --> "Lara Dutta"
Cardinal number prefix	"five coins" --> "coins"
Concept begins with dollar sign	"$200 Million" --> "dollar"
Subordinate clause without "that"	"the book Mary read" --> "book"

Thus, we define a "syntax-based concept node" (SBCN) to be the 4-tuple:

<name,{parents},{children},{tuples}>

where "name" refers to the noun phrase for the discourse entity for which the node is constructed, "{parents}" and "{children}" refer to the (possibly empty) sets of parent and children nodes for the concept, and "{tuples}" refers to the tuples in which the concept (discourse entity) appears.

We characterize this knowledge structure as a concept node because it encapsulates the knowledge for a single object of discourse, that is, a "concept." And we also describe this structure as "syntax-based" because the heart of the structure is the embedded set of proposition tuples, which are themselves syntax-based. We believe that this structure fully takes into account phrasal concepts and possesses an elegant instantiation mechanism that supports discourse analysis and provides the entry point into the relationship between the knowledge base and WordNet, which we employ as an essential part of the reasoning mechanism.

The Concept Network

From the definition of SBCNs in the previous section, we observe that a node may have more than one parent. For example, a node for "John Hancock" may have parent links to both

"insurance company" and "revolutionary war hero." Similarly, a node may have more than one child. For example, the concept "submarine" may have children "Russian submarine" and "attack submarine". Thus, in general, the set of concept nodes is organized into a network.

However, we observe also that some nodes or sets of nodes may be isolated. For example, if the text database contained only one sentence involving a "Russian submarine" and the sentence does not involve any discourse entities that appear in any other sentences (for example, "The Russian submarine sank."), then "Russian submarine" and its derived parent "submarine" would be connected to each other, but they would be isolated from the rest of the network. Therefore, SBCNs constructed in the manner described are organized into a network of one or more disjoint subnetworks, each containing one or more related concept nodes.

To facilitate question answering, when concepts are added to the network, "nicknames" are generated for proper nouns so that only one node is created for each proper noun concept, and each of its nicknames is mapped as an alias to the concept. If a node for any of the other nicknames already exists in the network, a new node is not created, and instead, the affected tuples are added to the existing node. Thus, for example, an attempt to create a new node for the concept "John Kennedy" may encounter the previously existing concept node for "John F. Kennedy", in which case the new proposition tuples would be added to the existing node. The concept addition logic is so constructed that for such proper noun associations, the longer concept is maintained as the concept node. If a shorter concept is encountered first, then when the related longer concept phrase is added to the network, it subsumes the shorter one by adjusting all necessary parent and child links and adding an alias for the shorter one linking it to the master concept node. In this manner, the concept network is constructed to contain only one instance of each proper noun concept.

The concept network is also linked to WordNet by a simple mechanism. If the concept name for a node appears in WordNet, then the analysis of "is-a" relationships may proceed from within the network and into WordNet by examination of the hypernym tree for the concept name. Thus, for example, the network may contain a concept node for "Russian submarine" and its derived concept "submarine', and since the latter appears in WordNet for which "vehicle" is one if its hypernyms, then one would be able to answer a question that relies on knowing that a Russian submarine is a vehicle. This linkage would exist for most concepts that are constructed for common nouns, and for many concepts whose derivations produce common nouns, and for such proper noun concepts as already appear in WordNet, such as "Nobel Prize".

It is significant that the concept network contains nodes only for concepts based on noun phrases and phrasal concepts, such as "berries" and "mammals that live in the seas". This approach is quite different from structures centered on verbs as the lexical objects around which thematic roles or slots are instantiated. In the network that we propose, the verbs appear as predicates in the proposition tuples that are indexed by the concept nodes.

CHAPTER THREE: QUESTION ANSWERING

Question answering is a three-step process in which: (a) the question is analyzed to produce question pattern tuples; (b) the proposition tuples of interest are retrieved from the concept network; and (c) the tuples retrieved are then examined to search for candidate answers. These steps are discussed separately in the sections below.

Question Analysis

We decompose questions into proposition tuples in the same manner as for the document set, with the addition of free variables for the desired answer. These variables may take the form of a general directive, such as "*who", "*what", *when", "*where", and "*why", or a target preposition type, such as "*in", when the answer is expected to be modified by the preposition. We also define the answer variable "*ans" to serve as a referent to a candidate answer obtained in response to a previous tuple, so that we can construct question patterns as Boolean combinations of separate tuple patterns.

For simple questions, a single tuple pattern may suffice. For example, for the question *"What did Peter eat?"* the corresponding question tuple is <'Peter','did eat',_, _,_,"*what">, where we have indicate the free variable with a leading asterisk and null values with underscores for clarity of display.

More complex questions require the Boolean conjunction of question pattern tuples. For the question *"What kind of car does Peter drive?"* the question decomposes into the conjunction:

<'Peter','does drive',_,_,_,"*what"> <and><'*ans','is',_,_,_,"car">

Peter may drive his mother crazy or a nail with a hammer, but neither is a "car" and will not be returned as an answer because the second pattern would not match in such circumstances.

In a similar fashion, some questions require disjunctive combinations of question pattern tuples to accommodate alternative syntactic realizations of the expected answer. For the question *"What is the title of the book?"* the question decomposes into:

<'*what','is',_,_,_,'the title of the book'> <or>
<'the title of the book','is',_,_,_,'*what'>

Still other questions may require both disjunctive and conjunctive patterns. For example, the question *"What kind of book did Peter give to Mary?"* the question decomposes into:

(<'Peter','did give',__,'to Mary,',__,*what'> <or>
'Peter','did give',__,_,Mary,'*what'>) <and>
<'*ans','is',_,_,_,'book'>

where the disjunction is needed to find answers where the document base contains sentences such as *"Peter gave Mary the novel."* and *"Peter gave the novel to Mary."*

The final step in question analysis involves creating passive construction question patterns for active construction patterns, and vice versa, as well as constructing possessive form questions patterns from question patterns in which subjects or direct objects include the preposition "of." Any patterns produced at this step are added disjunctively to the existing Boolean combination of question tuples.

Table 2 lists sample factoid questions from the TREC 2005 QA test set, together with the question patterns derived from them by the SEMEX tool, which is discussed in Chapter 4. These serve to illustrate the types of patterns that may be generated for typical factoid question forms and the ability of the proposition tuple format to express propositions. The reader should note

that the question pattern shown in the table is taken directly from the SEMEX graphical user interface and includes all components of each pattern, with null values shown as empty strings.

Table 2. Question Patterns for Sample TREC 2005 Factoid Questions.

No. Question / Question Pattern

66.1 When did the submarine sink?
<or> [S:the submarine][V:did sink][GI:][M:*when][IO:][DO:]

66.2 Who was the on-board commander of the submarine?
<or> [S:*who][V:was][GI:][M:][IO:] [DO:submarine on-board commander]
<or> [S:submarine on-board commander][V:was][GI:][M:][IO:][DO:*who]
<or> [S:the on-board commander of the submarine][V:was][GI:][M:][IO:][DO:*who]
<or> [S:*who][V:was][GI:][M:][IO:][DO:the on-board commander of the submarine]

66.4 How many crewmen were lost in the disaster?
<or> [S:*crewmen][V:were lost][GI:][M:in the disaster][IO:][DO:]
<or> [S:][V:lost][GI:][M:in the disaster][IO:][DO:*crewmen]

66.6 In what sea did the submarine sink?
<or> [S:the submarine][V:did sink][GI:][M:*in][IO:][DO:]
<and> [S:*ans][V:is][GI:][M:][IO:][DO:sea]

67.1 Who won the crown?
<or> [S:*who][V:won][GI:][M:][IO:][DO:the crown]
<or> [S:*who][V:was crown][GI:][M:][IO:][DO:]

67.2 What country did the winner represent?
<or> [S:the winner][V:did represent][GI:][M:][IO:][DO:*what]
<and> [S:*ans][V:is][GI:][M:][IO:][DO:country]

67.4 Where was the contest held?
<or> [S:the contest][V:was held][GI:][M:*where][IO:][DO:]
<or> [S:the contest][V:was][GI:][M:*where][IO:][DO:]

68.1 Where is Port Arthur?
<or> [S:Port Arthur][V:is][GI:][M:*where][IO:][DO:]

69.5 At what stadium was the game played?
<or> [S:the game][V:was played][GI:][M:*at][IO:][DO:]
<and> [S:*ans][V:is][GI:][M:][IO:][DO:stadium]

71.1 What type of plane is an F16?
<or> [S:an F16][V:is][GI:][M:][IO:][DO:*what]
<or> [S:*what][V:is][GI:][M:][IO:][DO:an F16]
<and> [S:*ans][V:is][GI:][M:][IO:][DO:plane]

87.2 When did Enrico Fermi die?
<or> [S:Enrico Fermi][V:did die][GI:][M:*when][IO:][DO:]

87.3 What Nobel Prize was Fermi awarded in 1938?
<or> [S:Fermi][V:was awarded][GI:][M:in 1938][IO:][DO:*what]
<and> [S:*ans][V:is][GI:][M:][IO:][DO:Nobel Prize]

87.5 What is Enrico Fermi most known for?
<or> [S:Enrico Fermi][V:is known][GI:][M:most;*for][IO:][DO:]

85.5 Why did the Grand Cayman turn away a NCL ship?
<or> [S:the Grand Cayman][V:did turn away][GI:][M:*why][IO:][DO:a NCL ship]

90.3 Which Virginia vineyard produces the most wine?
<or> [S:*which][V:produces][GI:][M:][IO:][DO:the most wine]
<or> [S:the most wine][V:produces][GI:][M:*by][IO:][DO:]
<and> [S:*ans][V:is][GI:][M:][IO:][DO:Virginia vineyard]

We observe that typical factoid and list type questions may be represented by a simpler grammar than input text in general, so that their decomposition may be achieved through relatively simple finite state automata that parse part-of-speech tagged questions into various phrase types (e.g., noun, verb, prepositional, infinitive). However, this may change as the boundaries of research in this area are extended to more difficult question formats. Nevertheless, independent of the means of decomposition, the question patterns so decomposed are processed against the concept database as described in the next section.

Tuples of Interest

The concept network is, by construction, indexed by the concept names it includes, so it is not necessary to conduct a blind search for proposition tuples that involve any of the discourse entities that are of interest in answering a particular question. In this manner, the network supports incremental growth as the body of knowledge is acquired, and there will not be a significant performance penalty for such growth, although storage requirements will, of course, increase.

The tuples of interest are retrieved as follows. First, noun phrases are extracted from the various components of the question tuple or tuples. This set is augmented with the WordNet hyponyms for each head noun in the set, and by the inclusion of the question target and its synonyms. Next, for each such noun phrase, the concept network is checked for the presence of a concept node by that phrase, or the presence of a node to which such phrase is mapped as an alias. If a concept node is found, then all tuples associated with that node are included in the return set. The child nodes of the concept node are also examined recursively, and any new tuples found are also added to the return set. Once all noun phrases are checked against the hierarchy, the resultant tuples are returned as the set of tuples of interest. By construction, every tuple in this set contains one or more of the phrases of interest, and there should not be any tuples in the concept network that contain one of the desired phrases which are not included in this set.

Question Pattern Matching

Question pattern matching proceeds by a straightforward unification algorithm in which the entire Boolean combination of question tuples is applied to each tuple until an answer is found, for a factoid question, or for a list question, until all tuples are examined.

The examination of a single proposition tuple proceeds by checking its tuple components against the non-null components of the question pattern. For each such component that does not involve a free (answer) variable, a matching algorithm is executed. If any such component fails to match, the proposition tuple is rejected and examination proceeds to the next tuple in line. For subjects and direct objects, an examination set is created consisting of all system aliases for a proper noun phrase, if any, otherwise the common noun phrase itself, augmented with the WordNet hyponyms for the head noun and by all aliases for the target if the phrase contains a member of the target synset. For example, if the target is "Russian submarine Kursk" and the proposition tuple contains the word "submarine", then all target aliases, including the word "Kursk" are included, so that a match will be found against a question pattern in which "Kursk" is specified. And where the question inquires about a "ceremony", the proposition tuple will be checked for matches against the hyponyms of that term, including, for example, "pageant."

For verbs, the main verb from the proposition tuple is extracted and WordNet methods are used to obtain all verb roots. Each root is then compared against the WordNet root of the main verb in the question tuple and a match is found if any root of one is found in any synset of the WordNet hypernym expansion of the other. This algorithm will find a match between the verbs "give" and "transfer", for example, since "transfer" appears in the hypernym tree for "give."

Gerunds and infinitives are matched in the same manner as main verbs, except that the infinitive “to do” in the question pattern is considered to match any infinitive that is present in the proposition tuple.

Indirect objects receive special treatment, since a nominal indirect is semantically equivalent to a similarly structured sentence in which the same noun phrase occurs as the object of a “to” or “for” preposition. Accordingly, the proposition tuple's nominal indirect is checked, if any, and if none, then its modifiers commencing with “in” or “for”, if any, are examined.

Modifiers also receive special treatment since there can be more than one in either the question pattern or the proposition tuple. Where the question pattern contains more than one modifier, a match is recorded only if all modifiers are matched independently. However, so long as all question modifiers are matched, it does not matter that a proposition tuple may have additional, unmatched modifiers, since these are, by construction, irrelevant to the question. We seek to achieve robustness in matching through examinations of WordNet synonyms for a noun phrase head, whenever possible, which obtains for most common nouns.

Where the question pattern component contains a target answer variable, an answer retrieval algorithm is executed according to the component type. For subjects and direct objects, the algorithm returns the corresponding component of the proposition tuple if the expected answer type is confirmed through WordNet. For example, a location type is confirmed if “structure” or “location” appears in the head noun's hypernym tree. Similarly, verbs and other tuple components are retrieved after appropriate type checks by algorithms that parallel their corresponding match methods.

If a match is not obtained for a given proposition tuple, then the reciprocal predicate is checked if the subject and direct object are both nonempty, there is no gerund/infinitive, and the

main verb is not copular, not the verb "do", and possesses an antonym in WordNet. This construction increases recall by supporting a match for the question "What did Mary receive?" where the database contains, for example, "Peter gave Mary the book." Once the reciprocal tuple is formed, it is matched in the same manner described above for the direct pattern.

Different processing is necessary for question patterns that seek to confirm "is-a" relationships for previously found candidate answers. Where, for example, we seek to confirm whether a candidate answer for the location of the sinking of the submarine Kursk is in fact an instance of the "sea" class, the parents of the candidate answer are searched recursively to the head of their equivalence hierarchy for the desired class. If none is found, then the WordNet hypernym tree for each root of the hierarchy is checked. We note that there can be more than one such root, since a concept node can have more than one parent. If the desired class name is found in any location check, then the candidate concept is deemed a member of the class and a match is recorded.

CHAPTER FOUR: TEST AND EVALUATION

The SEMEX Test Environment

Our SEMEX (SEMantic EXtractor) tool is a test bed environment for evaluating and refining semantic extraction and question answering algorithms. SEMEX provides the graphical user interface shown in Figure 1 for viewing the intermediate results at key stages of the knowledge extraction process.

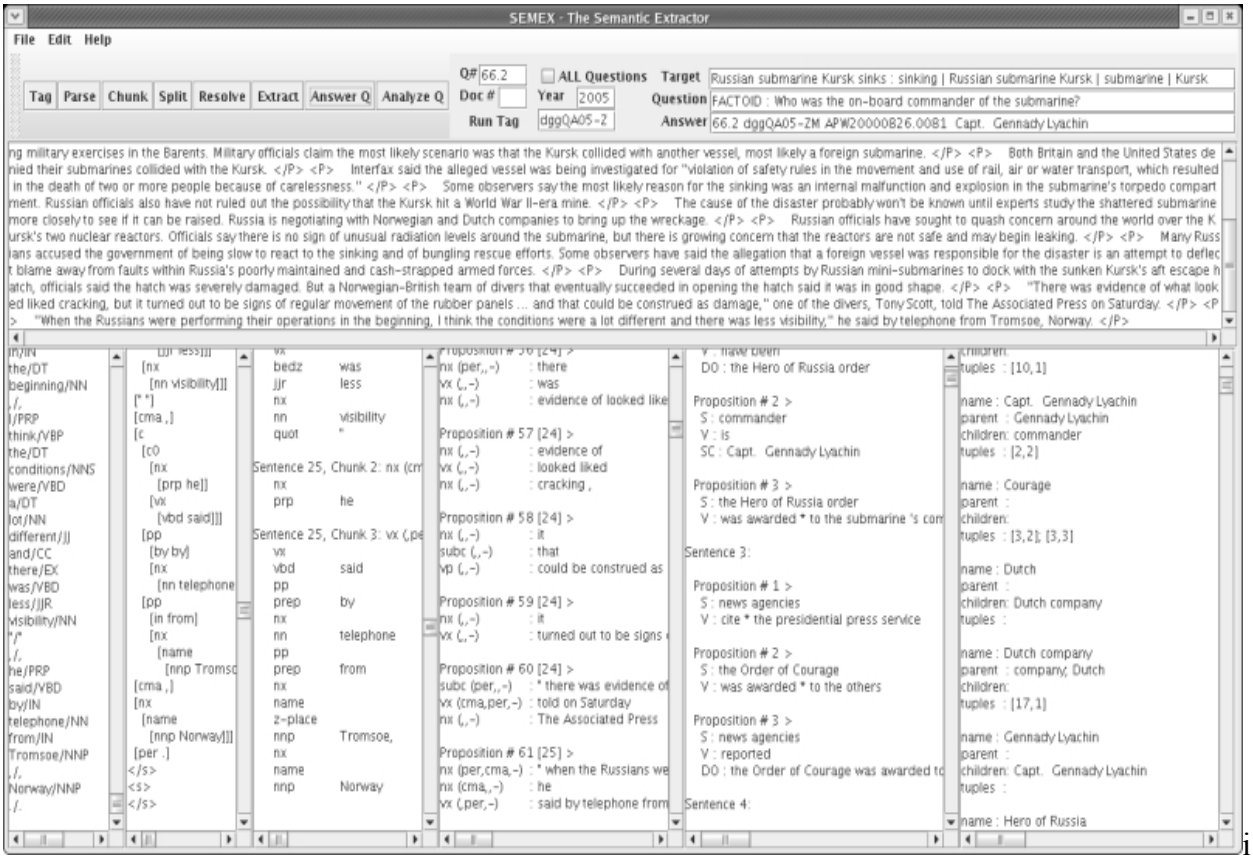

Figure 1. SEMEX Graphical User Interface

Figure 2 shows the top-level architecture of the SEMEX question answering tool. The tool creates a concept network by processing the input text through a cascade of modules for: (1) part of speech tagging; (2) partial parsing; (3) chunking; (4) sentence decomposition; (5) resolution; and (6) concept extraction.

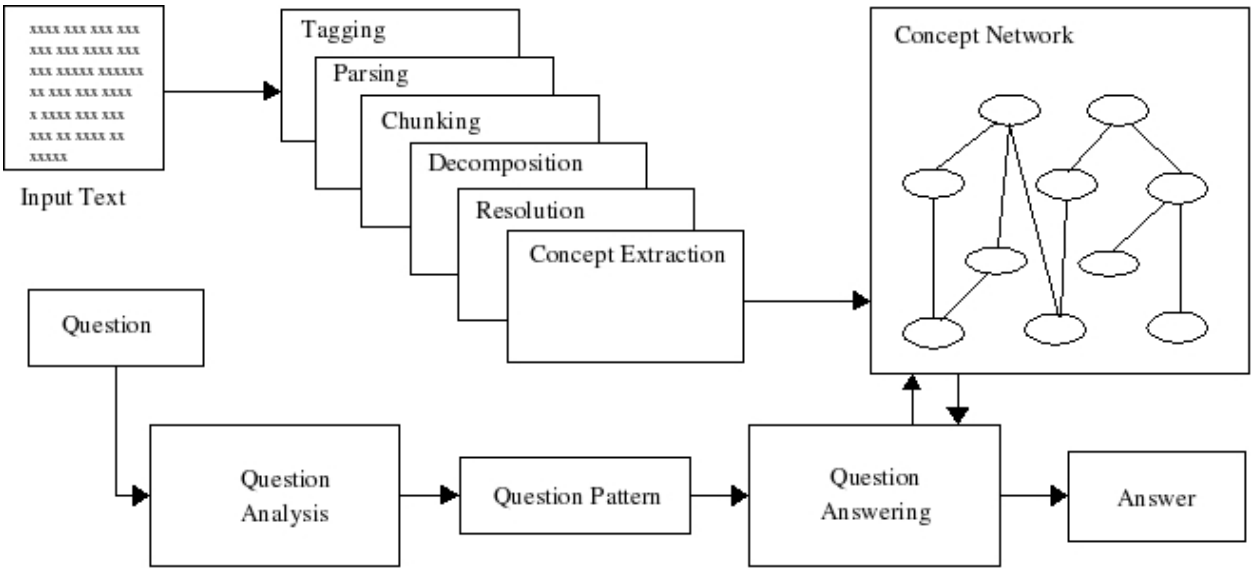

Figure 2. SEMEX Top-level Architecture

Prior to tagging, SEMEX removes spurious HTML escape codes, newswire datelines, paragraph tags, and similar cleanup items. Punctuation is also separated; dates, numbers and proper nouns are aggregated; and initials are flagged so the tagger does not interpret them as sentence boundaries. Tagging is performed by the Brill tagger (Brill 1994), whose output is corrected as needed.

Parsing is performed using the Cass partial parser (Abney 1996). SEMEX then applies its own comprehensive set of empirically derived heuristics to build up phrases at the chunking

stage. The resultant parse trees are then simplified and reorganized in the sentence decomposition stage to create separate propositions for the clauses, appositions and coordinations of complex sentences, a process based on extending the sentence splitting heuristic method described in (Glinos 1999). Syntactic roles are then assigned to the proposition components and pronoun references are resolved. And finally, proposition tuples are created from the resolved proposition, and the network of concepts is built from the discourse entities contained in the resolved propositions, as described in Chapter 3.

SEMEX performs question analysis as described above, utilizing separate finite state automata for each question type implemented to take advantage of the simpler grammar of typical factoid questions, thereby bypassing the need for parsing the tagged text. Instead, the tagged text is chunked manually into noun, verb, prepositional, adverbial, adjectival, infinitive, and gerund phrases, and the question patterns are built up incrementally from a single pass through the tagged question text. Currently, SEMEX implements the following question types: (i) who; (ii) what; (iii) what kind/type; (iv) when; (v) where; (vi) how; (vii) why; and (viii) which.

To extract the answer, SEMEX performs a unification of the question patterns with the relevant logical forms retrieved from the concept network and WordNet (Fellbaum 1998) is used in the unification process to improve recall, as outlined in Chapter 3.

The complete source code for SEMEX, consisting of over 48,000 lines of Java code, plus some C code, is contained in Appendix A to this dissertation. The files comprising this software package, and their basic functions, are shown in Table 3. To obtain a running configuration, one must also obtain separately and configure the Brill tagger, the Cass (Abney's) partial parser, WordNet 2.0, and the Java WordNet Library. The tagger code must also be modified very

slightly to write its output in a file, in addition to the screen, for use by downstream SEMEX modules.

Table 3. SEMEX Source Files and Functions

Source File	Principal Functions
semex.java	SEMEX graphical user interface
semexJNI.java, .c, .h	Java Native Interface code for integrating with the Brill tagger and the Cass partial parser
KstAnswer.java	question analysis and answering
KstChunk.java	classes to represent the chunks obtained from tagged text
KstConcept.java	concept node class methods
KstDictionary.java	interface to WordNet API
KstDiscourse.java	classes to resolve anaphora and assign syntactic roles
KstExtract.java	classes to extract proposition tuples from resolved forms
KstFile.java	TREC data file access methods
KstGroup.java	classes for chunking and grouping tagged text
KstSplit.java	classes to split chunked sentences into simpler forms
KstTuple.java	proposition tuple class methods
KstUtil.java	tagger correction and utility methods

Evaluation Against TREC Factoid Questions

The annual Text Retrieval Conferences (TRECs), co-sponsored by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense Advanced Research and Development Activity (ARDA), have been a focal point for leading edge question answering systems since 1999, when TREC first introduced its question answering track. (Voorhees 2005b) Many of the key researchers in the field have participated in TREC QA track evaluations.

The 2005 TREC QA Track continued the practice of recent years in using the AQUAINT Corpus of English News Text as the document set for evaluation. The AQUAINT corpus was produced by the Linguistics Data Consortium for the Advanced Question Answering for Intelligence (AQUAINT) program. This data set consists of 1,033,461 newswire articles from the English portion of the Xinhua News Service (People's Republic of China) (covering 1996–2000), the New York Times News Service (covering 1998-2000), and the Associated Press Worldstream News Service (covering 1998-2000). The articles comprise roughly 375 million words that correspond to approximately 3GB of data. The articles, which are in different formats for each data source, were selected to be representative of real world data, and so contain their full SGML markup and have not been processed to correct for typographical errors, misspellings, grammar mistakes, and extraneous characters. The newswire articles for a single day are contained in three files, one for each service, with multiple articles in each file.

The TREC 2005 QA question set consisted of 530 questions, of which 362 were "factoid" questions calling for a single item as a response; 93 questions were "list" questions calling for answers containing multiple items; and 75 were "other" questions, calling for information "nuggets" not contained in previous answers concerning the same target. The questions were organized into 75 groups, each related to a target concept, which was provided in the question set. The question set was provided by NIST as an XML-tagged file. Appendix B contains a version of that file with markup removed and formatted for clarity of display.

SEMEX was configured to exercise the TREC 2005 QA questions against the top-50 documents for each target returned by NIST's generic IR engine from the AQUAINT newswire collection. Since SEMEX does not possess an IR component, this was a necessity, although it was not guaranteed that the answers, if any, would be contained within the top-50 document lists.

Moreover, SEMEX did not implement all of the question types covered by the test set, for example, question 66.3, which asked, “The submarine was part of which Russian fleet?” Nor did SEMEX make use of the dates, titles, and datelines that were separately marked up in the documents, as these were considered TREC-specific features that are irrelevant to an evaluation of the formalism described in this thesis.

For these reasons, SEMEX output for the first 200 factoid questions was analyzed in detail and scored manually using the TREC-furnished factoid answer patterns. Appendix C contains the factoid answer patterns for the 2005 question set. The answer set reflects that factoid questions 77.5, 78.3, 81.6, 83.3, 90.3, 91.6, 110.3, 111.5, 125.7, and 126.5 did not have any known answers in the AQUAINT corpus, so that “NIL” was the correct response for those questions. Since the first two hundred factoid questions in the test set were used for analysis, regardless of type or content, the results reported reflect an objective test of the algorithms.

The mix of questions types for the first 200 factoid questions in the test set is shown in Table 4. The type “Prep” refers to questions that begin with a preposition, for example, “In what sea did the submarine sink?” The type “Other” refers to questions that do not begin with keywords for any of the other categories, for example, “The submarine was part of which Russian fleet?” SEMEX implemented many variations for all question types except for the “Other” category, but not all variations were covered, as discovered during analysis of the test results. SEMEX did not implement any forms for the “Other” category. In both cases, this was due to limitations in development time for the tool, not limitations of the formalism.

Table 4. Question Types for First 200 Factoid Questions from 2005 TREC QA Test Set

Question Type	Occurrences
Who	33
What	66
What kind/type	2
Where	22
When	18
Why	1
How	35
Which	4
Prep	11
Other	8

To eliminate the limitations of the tool and the document set, and so to obtain a more accurate appraisal of algorithm performance, the raw results were analyzed as follows.

First, a question was disregarded if: (a) its answer was not in the top-50 documents; (b) the document containing the answer was not readable by SEMEX; (c) the answer was in a direct quote; (d) the answer was in the dateline or headline; (e) the question form was not one of the types implemented in SEMEX; or (f) the answer key provided a wrong answer, or no answer, to the question, where NIL was not the correct answer. All of these were considered "non-starter" situations where the algorithms did not have the opportunity to perform.

Categories "a", "e", and "f" are self-explanatory. With respect to "b", there was generally poor parsing performance against the newswire articles that comprised the document set, as they were characterized by numerous spurious HTML codes and meta-data within the text, such as identification of sources, sports scores, bulletized lists, instructions in the form of incomplete sentences, and by the flowery and run-on sentences that characterize much news reporting. Moreover, the current SEMEX implementation could not read some documents at all,

particularly those with multiple embedded datelines and sports scores, and some of these documents were the ones that contained the answers.

With respect to "c", there were a few instances where the answer could be found only in a direct quote. However, the current SEMEX implementation treats such content as hearsay in the legal sense and does not attempt to construct proposition tuples from the contents of directly quoted material, although the fact that the statement is made is extracted. For example, if the text reported "Peter said, 'Paul is a thief,'" SEMEX would extract the proposition that Peter said those words about Paul, but not that Paul is a thief.

The most interesting exclusion is "d". In TREC, the participants are encouraged to use the SGML markup in their information retrieval and question answering. Thus, if newswire article reported that something occurred "today", then in the TREC context, it is appropriate to ask when the event occurred and to expect the date of the article to be returned as the answer. But the algorithms described in this dissertation and implemented in SEMEX are intended to be fully general and not tied to date-tagged documents. For our application, using headlines and datelines would be a "crutch" that would artificially enhance the algorithms' ability to extract concepts and relationships from the text, which is their basis.

Conversely, we considered an answer correct if the answer made sense in the context of the document in which it was found, even though the answer was not what is specified in the TREC-furnished answer key. For example, for Question 66.1, the answer in the key is "(Aug(.|ust)? 12,?)?2000", while the answer found by SEMEX was "Aug. 12". We nevertheless accepted the answer as correct because the year did not appear in the document, except in the date of the newswire article.

We also accepted a correct answer if manual corrections to the chunked output in the SEMEX GUI resulted in the correct answer being found by SEMEX. The rationale for this is that this is clearly a shortcoming of the parsing front of SEMEX, not of the concept extraction. These situations occurred primarily in the recognition of nested subordinate and relative clauses, and infinitive phrases. An answer was also considered to have been found successfully if only minor manual corrections to the parsed output of the document containing the answer or a substitution of a term in question was sufficient to allow SEMEX to find the correct answer, as these situations were considered shortcomings of document and question parsing, and not indicative of algorithm functional performance. Similarly, an answer was accepted if it could be found by SEMEX by substituting a word or two in the question, usually referring to the target, as this was considered a deficiency in anaphora resolution of the question or a poorly worded question. For example, for the target "Miss Universe 2000 crowned", question 67.4 asked, "Where was the contest held?" SEMEX did not find the answer for the question in this form, but when "contest" was replaced with "ceremony", the answer was found because in WordNet a "pageant", such as the 49th Miss Universe Pageant, is not a "contest", but the hypernym tree for "pageant" contains the word "ceremony."

Importantly, however, questions whose answers were not found by SEMEX and which required reasoning over the concept set or more than minor syntactical parsing adjustments or question substitutions were retained as failures, as these indicated areas of further inquiry.

Table 5 presents SEMEX performance against the first 200 factoid questions analyzed in line with the considerations above. Giving SEMEX credit for the answers found with reparses and substitutions, and discarding the fifth through eleventh categories in the table for the reasons cited above, SEMEX achieved a raw score of 20.59% for 28 correct answers and an adjusted

score of 50.74% for 69 correct answers out of the 136 non-discarded questions. Both of these scores compare favorably with the performance of other systems. For TREC 2004, (Voorhees 2005b) reports a correct answer accuracy range of 21.3% to 77.0% against the factoid questions for the top ten performing systems, with only the top 3 systems achieved factoid accuracy scores greater than 35%. With sixty-three runs submitted to the QA track, it is clear that most of the systems achieved factoid accuracies below the lower bound of this range.

Table 5. SEMEX Results for First 200 Factoid Questions from 2005 TREC QA Test Set

Response Category	Responses
Answer found on initial pass	28
Answer found on reparse	23
Answer found with substitution	10
Answer found with both parse and substitution	8
Answer not in Top Docs for question	29
Unable to read document containing answer	8
Answer in direct quote	2
Answer in headline or dateline	12
Question type not covered	6
Question parsed incorrectly	5
Wrong or no answer furnished in answer key	2
Answer not found	67
False positives	45
Adjusted false positives	29

For the same 200 questions, SEMEX also produced incorrect answers for 45 questions, for a raw false positive or precision value of 22.5%. However, this value reflects the same difficulties in parsing the document set as above. An examination of the individual questions where false positives were produced reveals that sixteen such responses were for questions in which the correct answer could be found by a reparse or substitution. Discounting the false

positives for these questions, because a correct answer and a false positive are mutually exclusive, results in an adjusted false positive or precision value of 14.5%.

Beyond these encouraging results, detailed investigation into the operation of the proposed concept extraction and question answering algorithms reveals that the data structures and algorithms do indeed perform as designed.

Algorithm operation is illustrated by SEMEX performance against TREC question 66.7, which asked the list question, *"Which U.S. submarines were reportedly in the area?"* for the target *"Russian submarine Kursk sinks"*. For this question, SEMEX generated the question patterns:

*<or> [S:*which][V:were][GI:][M:reportedly;in the area][IO:][DO:]*
*<and> [S:*ans][V:is][GI:][M:][IO:][DO:U.S. submarine],*

and returned the answer "the Toledo". What is significant about this result is that the text from which the answer was obtained consisted of a document whose body contained only these sentences:

<P> The second U.S. submarine in the Barents Sea when the Kursk sank was the Toledo, a Russian news agency reported Thursday. </P> <P> The agency, Interfax, said the Toledo was in the area along with another U.S. submarine, the Memphis, during the Russian naval exercises in mid-August, when the Kursk sank, with the loss of 118 lives. </P>

In this example, the fact that the Toledo was a U.S. submarine was extracted from the first sentence, while the fact that it was reportedly in the area when the Kursk sank was extracted from the second sentence. The connection between these two facts was provided by the concept network, in which the "Toledo" concept had the parent "second U.S. submarine in the Barents Sea when the Kursk" from a copular proposition, from which SEMEX derived the parent "second U.S. submarine", which by a further derivation had "U.S. submarine" as parent, which was the desired class. We note that SEMEX was unable to find the second valid answer, "the

Memphis", since SEMEX did not understand that vessel to have been in the area, although its ancestor chain revealed that SEMEX did recognize it as a U.S. submarine.

Against the same target, SEMEX was able to find the correct answer, "Capt. Gennady Lyachin," in response to question 66.2, a result which only 2 out of 71 total TREC QA runs were able to achieve. The factoid question involved was *"Who was the on-board commander of the submarine?"* The relevant input sentence was *"The Hero of Russia order, one of the country's highest honors, was awarded to the submarine's commander, Capt. Gennady Lyachin."* SEMEX was able to find the answer in this case because (a) the SEMEX sentence decomposition logic identified the apposition and created the separate copular proposition "Capt. Gennady Lyachin" is-a "commander", and (b) SEMEX's question answering logic, discussed above, was able to associate "the submarine's commander" with "the on-board commander of the submarine."

In another example, for the target *"Rose Crumb",* question 120.3 asked *"What organization did she found?"* For this question, SEMEX returned the correct answer: *"the volunteer Hospice of Clallam County,"* which only 7 of 71 runs were able to answer correctly. Examining SEMEX operation in this case revealed that the relevant input sentence was:

> Crumb, who founded the volunteer Hospice of Clallam County, Wash., in 1978, said she has learned ``courage, patience and acceptance'' by working with families to provide their dying relative with a dignified end.

SEMEX was able to identify the relative clause and create a separate proposition for it, which was easily matched against the question because "Hospice of Clallam County" was recognized as a business organization. SEMEX was able to split off the relative clause and create a separate proposition for it. That proposition, as shown in SEMEX's resolution window, was:

Proposition # 5 >
S : Crumb

V : founded
DO : the volunteer Hospice of Clallam County

which was easily matched against the question because "Hospice of Clallam County" was recognized as a business organization.

Question 87.3 asked the factoid question, *"What Nobel Prize was Fermi awarded in 1938?"* for the target *"Enrico Fermi."* For this question, SEMEX correctly returned *"the Nobel Prize for Physics"*, which only 10 of 71 runs achieved. This was made possible by the parent-child relationship present in the concept network between "Nobel Prize" and the answer returned. Indeed, a separate test has shown that SEMEX would still have returned the correct answer if the question had asked what "prize" had Fermi been awarded, again as a consequence of the concept network. Moreover, the link from the concept network to WordNet is illustrated by a further test that asked what "award" had Fermi been awarded, which succeeded even though "award" was not an ancestor of the answer in the concept network, but because it was nevertheless a WordNet hypernym of the ancestor "Nobel Prize."

SEMEX performance in these investigations confirmed that there were questions that WordNet would consider inaccurately stated. For example, for the target *"Miss Universe 2000 crowned,"* factoid question 67.4 asked, *"Where was the contest held?"* For this question, SEMEX was unable to find an answer, even though it had correctly split off the following proposition:

Proposition # 2 >
S : Miss India Lara Dutta
V : was crowned * during the 49th Miss Universe pageant
* in Nicosia, Cyprus * early Saturday
DO : Miss Universe 2000

Examining SEMEX's internal operation for this question revealed that the reason the location adverbial *"in Nicosia, Cyprus"* was not matched is because a "pageant" does not have "contest" in any of its WordNet synsets. However, as noted above, when "contest" is replaced by "ceremony" in the question, the correct answer was found.

CHAPTER FIVE: CONCLUSIONS

Concept Extraction and Question Answering

Considering as a whole the experimental results reviewed above, we believe that they confirm the value of question answering through pattern matching at an intermediate level of representation, one that is abstracted from the surface syntactical patterns where much of the previous work on pattern matching has focused. For it is at the abstract level where we may take advantage of the fact that while there are many ways to say the same thing, it is still one thing that is being said. By answering questions at this level, it is possible to obtain a certain level of invariance in performance when confronted with syntactic variations but semantic invariance in either the questions or documents. And this, we feel, is as it should be, since we do not wish our automated question answering systems to require that the question or the documents be expressed in only limited ways.

Moreover, we believe that our results confirm the value of decomposing or splitting sentences into the simplest possible clauses without changing meaning. The value of doing so is that it permits us to base our logical forms, the proposition tuples, on the syntactic form of a normalized atomic proposition, by which we mean a proposition involving a single predicate whose arguments are the non-null syntactic components of such a clause.

The value of using WordNet is also confirmed by our results. By using this lexical resource's synonym, hypernym, and antonym relations, the algorithms we propose are able to find answers without requiring exact matches in wording. In our implementation, WordNet was

also used to assist tagger correction algorithms and to classify object by types sych as location, quantity, and similar categories. In this manner, we obtained robustness in question answering by using WordNet.

Our results also confirm the value of deriving parent concepts from child concepts, as this provides the concept network with many necessary concepts that one cannot reasonably expect to be set forth explicitly in source document text.

Finally, our results confirm the value of a concept network of syntax-based concept nodes, constructed as described, as an indexing mechanism for the logical forms that constitute the knowledge base. We were thus able to find the proposition tuples of interest without the need to search the entire knowledge base. We hypothesize that by using the concept nodes as an indexing mechanism in this manner, this approach can support the incremental acquisition of a knowledge base without incurring a prohibitive performance penalty for question answering when the knowledge base becomes very large.

From an implementation standpoint, we observe that parsing performance, for both source text and question, is crucial for algorithm performance. This is not surprising, for our approach requires understanding the syntax so that the semantics may be inferred.

Limitations of Approach

Although the implementation of our approach was limited to factoid question answering, we argue here that proposition tuples, extracted as we have described and embedded within a network of syntax-based concept nodes, support more advanced reasoning. We can observe the general contours of how to approach deeper reasoning from the manner in which complex

factoid questions are answered. As we have seen, some questions decompose into Boolean combinations of two or more question patterns, each of which could be satisfied by a different proposition tuple or by an inference from the parent-child relationships encoded into the structure of the concept network. By straightforward extension, we can envision how a reasoning component, perhaps a theorem prover or a resolution module, can generate a set of candidate hypotheses, each of which is a Boolean combination of question patterns. Then, if any of the pattern sets is satisfied by the tuples of interest in the network, then the hypothesis would be considered satisfied and appropriate action taken. Thus, for example, if a question were "Is Socrates mortal?" a reasoning component might generate alternative hypothesis patterns for (a) Socrates is mortal; and (b) Socrates is a man AND all men are mortal. In this example, hypothesis (a) could be tested as a single question pattern, while hypothesis (b) could be tested against the concept base in the same manner as a multi-part factoid question, as discussed previously in this dissertation.

One may inquire whether the language of Boolean combinations of proposition or question patterns is sufficiently expressive to cover the range of input propositions or inquiry hypotheses that one may encounter. We argue that it does, for the reason that our proposition tuples span the possible components of grammatically well-formed statements or questions, and because the statements and questions of interest to this line of research must be understandable, and therefore, well-formed.

The above notwithstanding, there are, however, distinct constraints that restrict the application of our structures and methods. As our experience with newswire documents for TREC has demonstrated, the ability to acquire useful knowledge from text by extracting propositions and concepts as we have described depends strongly on the quality of the part of

speech tagging, parsing, chunking, and decomposition of the input text documents. This depends partly on the technologies employed, but also on the grammatical complexity and "noise" of the document domain. We observe in this regard, for example, that the domains of scholarly tracts, encyclopedic texts, medical journals, and newswire articles each presents different challenges, for which different techniques may need to be explored as these algorithms are applied in the future to different domains. Despite these variations, we anticipate that the concept extraction approach will remain the same, namely, obtaining high quality POS tagging and parsing to support downstream sentence splitting into atomic propositions, from which proposition tuples will be extracted as logical forms and embedded into a concept network.

We observe also that our approach does not expressly take into account word sense disambiguation. As a result, we anticipate that proposition tuples involving different senses of the same word, for example, the "bank" of a river and the "bank" that is a financial institution, will be indexed within the same concept node. The question for further investigation is whether the constellations of other syntactical components within the proposition tuples for each of the senses, will serve as selectional restrictions, so that question patterns pertaining to one sense will not match against tuples for another sense. No doubt degenerate examples can be constructed where such a question will obtain a mismatch, but what interests us is what will occur in practical domains. Nevertheless, the analysis and processing steps that we propose do leave room for incorporating a disambiguation module in a practical architecture, such as SEMEX, implementing our approach.

Finally, we observe that the robustness in question answering that is achieved by using WordNet must be tempered by the limitations of its linguistic formalism. For example, WordNet does not recognize that a beauty pageant as a "contest", although that association occurs in

common parlance. Instead, WordNet considers a pageant more properly as a "ceremony", which is, strictly speaking, correct. Nevertheless, much communication employs the "looser" conventions of popular discourse, and it is unclear how one may know this information and how it may be used effectively, recognizing that common usage is often inconsistent.

Future Research

Two distinct lines of further research suggest themselves based on the research reported above. The first involves continuing to develop a better TREC competitor. For this line, we consider essential the addition of an information retrieval (IR) component so that it will not be necessary to rely on the generic NIST search engine to retrieve documents for analysis. There are a number of available document search engines that are available in the public domain, such as the Lucene search engine. We believe that an IR component to SEMEX would be able to find more relevant documents and also to filter out some of the more difficult to parse documents, such as bulleted lists, multiple by-line documents, and other documents containing non-sentential content.

Whether or not an IR component is incorporated, a better TREC competitor would include improved input text file noise filtering so that SEMEX can read what is readable in every document that is processed. Also, we would expect improvements throughout the front end processing stages: tagging, parsing, chunking, and sentence decomposition. For TREC specifically, making use of headlines and datelines is essential, as shown by our experience. We would also expect to see more parent-child concept derivations and better anaphora resolution, including the resolution of discourse definite references. More question types must be

implemented, so that finding an answer does not fail because the question is not well understood. Finally, we would also add a component to filter nonsensical candidate answers, since processing stages through decomposition will not always succeed in generating well-formed propositions.

The second line of investigation involves moving towards a true "intelligent digital assistant" by using the algorithms described here to support the incremental acquisition of a knowledge base and methods for querying it effectively. In particular, we are intrigued with the prospect of developing algorithms for using the concept network as a knowledge base for answering questions that require reasoning about the concepts in the network, from quantitative and temporal aspects to hypothesis testing. Extending the present approach to reason about existentially quantified concepts, for example, would be straightforward, as the SEMEX implementation already recognizes such quantifiers and creates separate concepts for such recognized instances of universally quantified discourse entities. Additional work is no doubt needed to implement a reasoning component, but we argue that a syntax-based concept network will support such reasoning.

However, in this regime, it will be necessary to investigate additional aspects not addressed in the current research, such as maintaining the consistency of the knowledge base and how to recognize inconsistent input data. Nevertheless, with the addition of multimodal front and back end user interfaces, such as speech recognition and voice synthesis, one can imagine an intelligent digital assistant serving as one's personal secretary, reading and analyzing documents that one does not have time for, answering questions or summarizing what was read, and thus serving as the cumulative repository or knowledge base of one's personal, lifetime experiences.

APPENDIX A: SEMEX SOURCE CODE

The SEMEX tool that was used for this research was hosted on a Dell Inspiron 6000 laptop computer with a 1.6 GHz processor, 1 GB main memory, and a Linux file system partition of 40 GB. The program was developed and ran in a Fedora Core 3 Linux environment, running the software described below.

The source code for SEMEX comprises over 48,000 lines of Java code, plus a few lines of C code to link the tool to the Brill tagger and the Cass (Abney's) partial parser. The source code is contained in the following fifteen files, which can be viewed by selecting the file names, which link to the separate documents that contain them:

KstAnswer.java
KstChunk.java
KstConcept.java
KstDictionary.java
KstDiscourse.java
KstExtract.java
KstFile.java
KstGroup.java
KstSplit.java
KstTuple.java
KstUtil.java
semex.java
semexJNI.java
semexJNI.c
semexJNI.h

To obtain a running configuration, it is also necessary to obtain and install the following additional software: (a) Brill's tagger; (b) the Cass (Abney's) partial parser; (c) WordNet 2.0; (d) the Java WordNet Library (JWNL), version 1.3, and (e) Java SDK version 1.5. It is also necessary to make a minor modification to Brill's tagger so that it prints output to a file in addition to the screen.

APPENDIX B: TREC 2005 QA QUESTIONS

Target 66 : Russian submarine Kursk sinks

66.1	FACTOID	When did the submarine sink?
66.2	FACTOID	Who was the on-board commander of the submarine?
66.3	FACTOID	The submarine was part of which Russian fleet?
66.4	FACTOID	How many crewmen were lost in the disaster?
66.5	LIST	Which countries expressed regret about the loss?
66.6	FACTOID	In what sea did the submarine sink?
66.7	LIST	Which U.S. submarines were reportedly in the area?
66.8	OTHER	Other

Target 67 : Miss Universe 2000 crowned

67.1	FACTOID	Who won the crown?
67.2	FACTOID	What country did the winner represent?
67.3	FACTOID	How many competitors did the winner have?
67.4	FACTOID	Where was the contest held?
67.5	FACTOID	What was the scheduled date of the contest?
67.6	LIST	Name other contestants.
67.7	OTHER	Other

Target 68 : Port Arthur Massacre

68.1	FACTOID	Where is Port Arthur?
68.2	FACTOID	When did the massacre occur?
68.3	FACTOID	What was the final death toll of the massacre?
68.4	FACTOID	Who was the killer?
68.5	FACTOID	What was the killer's nationality?
68.6	LIST	What were the names of the victims?
68.7	LIST	What were the nationalities of the victims?
68.8	OTHER	Other

Target 69 : France wins World Cup in soccer

69.1	FACTOID	When did France win the World Cup?
69.2	FACTOID	Who did France beat for the World Cup?
69.3	FACTOID	What was the final score?
69.4	FACTOID	What was the nickname for the French team?
69.5	FACTOID	At what stadium was the game played?
69.6	FACTOID	Who was the coach of the French team?
69.7	LIST	Name players on the French team.
69.8	OTHER	Other

Target 70 : Plane clips cable wires in Italian resort

70.1	FACTOID	When did the accident occur?
70.2	FACTOID	Where in Italy did the accident occur?
70.3	FACTOID	How many people were killed?
70.4	FACTOID	What was the affiliation of the plane?
70.5	FACTOID	What was the name of the pilot?

70.6 FACTOID What was the outcome of the U.S. trial against the pilot?
70.7 LIST Who were on-ground witnesses to the accident?
70.8 OTHER Other

Target 71 : F16
71.1 FACTOID What type of plane is an F16?
71.2 FACTOID How fast can it fly?
71.3 FACTOID Who manufactures the F16?
71.4 FACTOID Where is this company based?
71.5 FACTOID Who manufactures engines for the F16?
71.6 LIST What countries besides U.S. fly F16s?
71.7 OTHER Other

Target 72 : Bollywood
72.1 FACTOID Where is Bollywood located?
72.2 FACTOID From what foreign city did Bollywood derive its name?
72.3 FACTOID What is the Bollywood equivalent of Beverly Hills?
72.4 FACTOID What is Bollywood's equivalent of the Oscars?
72.5 FACTOID Where does Bollywood rank in the world's film industries?
72.6 LIST Who are some of the Bollywood stars?
72.7 OTHER Other

Target 73 : Viagra
73.1 FACTOID Viagra is prescribed for what problem?
73.2 FACTOID Who manufactures Viagra?
73.3 FACTOID Who approved its use in China?
73.4 FACTOID What is the scientific name for Viagra?
73.5 FACTOID When did Viagra go on the market?
73.6 LIST In what countries could Viagra be obtained on the black market?
73.7 OTHER Other

Target 74 : DePauw University
74.1 FACTOID What type of school is DePauw?
74.2 FACTOID Where is DePauw located?
74.3 FACTOID When was DePauw founded?
74.4 FACTOID Who was president of DePauw in 1999?
74.5 FACTOID What was the approximate number of students attending in 1999?
74.6 LIST Name graduates of the university.
74.7 OTHER Other

Target 75 : Merck & Co.
75.1 FACTOID Where is the company headquartered?
75.2 FACTOID What does the company make?
75.3 FACTOID What is their symbol on the New York Stock Exchange?
75.4 FACTOID What is the company's web address?

75.5 LIST Name companies that are business competitors.
75.6 FACTOID Who was a chairman of the company in 1996?
75.7 LIST Name products manufactured by Merck.
75.8 OTHER Other

Target 76 : Bing Crosby
76.1 FACTOID What was his profession?
76.2 FACTOID For which movie did he win an Academy Award?
76.3 FACTOID What was his nickname?
76.4 FACTOID What is the title of his all-time best-selling record?
76.5 FACTOID He is an alumnus of which university?
76.6 FACTOID How old was Crosby when he died?
76.7 LIST What movies was he in?
76.8 OTHER Other

Target 77 : George Foreman
77.1 FACTOID When was George Foreman born?
77.2 FACTOID Where was George Foreman born?
77.3 FACTOID When did George Foreman first become world heavyweight boxing champion?
77.4 FACTOID Who did Foreman defeat for his first heavyweight championship?
77.5 FACTOID How old was Foreman when he first won the heavyweight championship?
77.6 LIST Name opponents who Foreman defeated.
77.7 LIST Name opponents who defeated Foreman.
77.8 OTHER Other

Target 78 : Akira Kurosawa
78.1 FACTOID When did Kurosawa die?
78.2 FACTOID When was he born?
78.3 FACTOID Which university did he graduate from?
78.4 FACTOID What was his profession?
78.5 FACTOID What was his English nickname?
78.6 FACTOID What was his wife's profession?
78.7 LIST What were some of his Japanese film titles?
78.8 OTHER Other

Target 79 : Kip Kinkel school shooting
79.1 FACTOID When did the school shooting occur?
79.2 FACTOID How many students were wounded?
79.3 LIST List students who were shot by Kip Kinkel.
79.4 FACTOID How many students did he kill?
79.5 FACTOID How old was Kip Kinkel when the shooting took place?
79.6 FACTOID How many bombs did investigators find in Kip's home?
79.7 OTHER Other

Target 80 : Crash of EgyptAir Flight 990

80.1	FACTOID	Where in the Atlantic Ocean did Flight 990 crash?
80.2	FACTOID	Who was the pilot of Flight 990?
80.3	FACTOID	Who was the co-pilot of Flight 990?
80.4	FACTOID	How many crew members were aboard?
80.5	FACTOID	How many passengers were aboard Flight 990?
80.6	LIST	Identify the nationalities of passengers on Flight 990.
80.7	OTHER	Other

Target 81 : Preakness 1998

81.1	FACTOID	Name the horse that won the Preakness in 1998?
81.2	LIST	List other horses who won the Kentucky Derby and Preakness but not the Belmont.
81.3	FACTOID	Who is the trainer of the Preakness 1998 winner?
81.4	FACTOID	Who finished second to the Preakness winner in 1998?
81.5	FACTOID	What was the track attendance for the 1998 Preakness?
81.6	FACTOID	What time did the race begin?
81.7	OTHER	Other

Target 82 : Howdy Doody Show

82.1	FACTOID	What year did the "Howdy Doody Show" first run on television?
82.2	FACTOID	On what date did the show go off the air?
82.3	LIST	Name the various puppets used in the "Howdy Doody Show".
82.4	LIST	Name the characters in the show.
82.5	FACTOID	The main puppet character was based on what person?
82.6	OTHER	Other

Target 83 : Louvre Museum

83.1	FACTOID	What was the Louvre Museum before it was a museum?
83.2	FACTOID	When was the Louvre transformed into a museum?
83.3	FACTOID	How many paintings are on permanent exhibit at the Louvre?
83.4	LIST	Name the works of art that have been stolen from the Louvre.
83.5	FACTOID	How many people visit the Louvre each year?
83.6	FACTOID	Who is president/director of the Louvre Museum?
83.7	OTHER	Other

Target 84 : meteorites

84.1	FACTOID	What is the largest meteorite found in the U.S.?
84.2	FACTOID	How heavy is it?
84.3	FACTOID	What is it called by the Indians?
84.4	FACTOID	Where is the world's largest meteorite?
84.5	FACTOID	How heavy is the world's largest meteorite?
84.6	FACTOID	How many metric tons of meteorites fall to the earth each year?
84.7	LIST	Provide a list of names or identifications given to meteorites.
84.8	OTHER	Other

Target 85 : Norwegian Cruise Lines (NCL)

85.1 LIST Name the ships of the NCL.
85.2 FACTOID What cruise line attempted to take over NCL in December 1999?
85.3 FACTOID What is the name of the NCL's own private island?
85.4 FACTOID How does NCL rank in size with other cruise lines?
85.5 FACTOID Why did the Grand Cayman turn away a NCL ship?
85.6 LIST Name so-called theme cruises promoted by NCL.
85.7 OTHER Other

Target 86 : Sani Abacha

86.1 FACTOID Give the month and year that General Abacha had a successful coup in Nigeria.
86.2 FACTOID What reportedly caused the death of Sani Abacha?
86.3 FACTOID How old was Sani Abacha when he died?
86.4 FACTOID Who was sworn in to replace Sani Abacha?
86.5 LIST Name the children of Sani Abacha.
86.6 OTHER Other

Target 87 : Enrico Fermi

87.1 FACTOID When was Enrico Fermi born?
87.2 FACTOID When did Enrico Fermi die?
87.3 FACTOID What Nobel Prize was Fermi awarded in 1938?
87.4 LIST List things named in honor of Enrico Fermi.
87.5 FACTOID What is Enrico Fermi most known for?
87.6 FACTOID Give the name and symbol for the chemical element named after Enrico Fermi.
87.7 FACTOID What country did Enrico Fermi come from originally?
87.8 OTHER Other

Target 88 : United Parcel Service (UPS)

88.1 FACTOID Where is UPS headquarters located?
88.2 FACTOID Who is the CEO of UPS?
88.3 FACTOID When was UPS's first public stock offering?
88.4 LIST In what foreign countries does the UPS operate?
88.5 FACTOID What color are UPS trucks?
88.6 FACTOID How much money did UPS pay out in insurance claims in 1984?
88.7 OTHER Other

Target 89 : Little League Baseball

89.1 FACTOID Where is the Little League World Championship played?
89.2 FACTOID On what street are the fields where the Little League World Series is played?
89.3 LIST What Little League teams have won the World Series?
89.4 FACTOID How many girls have played in the Little League World Series?

89.5 FACTOID What year was the first Little League World Series played?
89.6 FACTOID What is Little League Baseball's URL on the Internet?
89.7 OTHER Other

Target 90 : Virginia wine
90.1 LIST What grape varieties are Virginia wines made from?
90.2 FACTOID Approximately how many acres of grapes are grown in Virginia?
90.3 FACTOID Which Virginia vineyard produces the most wine?
90.4 FACTOID Who was Virginia's first and most famous wine maker?
90.5 LIST Name the Virginia wine festivals.
90.6 FACTOID Who was the former CEO who became a Virginia wine maker?
90.7 OTHER Other

Target 91 : Cliffs Notes
91.1 FACTOID Who originated Cliffs Notes?
91.2 FACTOID Whose works were the subject of the first Cliffs Notes?
91.3 LIST Give the titles of Cliffs Notes Condensed Classics.
91.4 FACTOID What company now owns Cliffs Notes?
91.5 FACTOID How many copies of Cliffs Notes are sold annually?
91.6 FACTOID What percentage of Americans have used Cliffs Notes?
91.7 OTHER Other

Target 92 : Arnold Palmer
92.1 FACTOID How many times did Arnold win the Masters?
92.2 FACTOID How many times did Arnold win the British Open?
92.3 LIST What players has Arnold competed against in the Skins Games?
92.4 LIST Which golf courses were designed by Arnold?
92.5 FACTOID What major championship did Arnold never win?
92.6 FACTOID What was Arnold's wife's first name?
92.7 OTHER Other

Target 93 : first 2000 Bush-Gore presidential debate
93.1 FACTOID Who moderated the first 2000 presidential debate?
93.2 FACTOID How long was the debate scheduled to be?
93.3 FACTOID On what university campus was the first debate held?
93.4 FACTOID Which major network decided not to televise the debate?
93.5 FACTOID In what state did Al Gore prepare for the first debate?
93.6 FACTOID On what date was the first debate?
93.7 LIST Who helped the candidates prepare?
93.8 OTHER Other

Target 94 : 1998 indictment and trial of Susan McDougal
94.1 FACTOID Who was Mrs. McDougal's lawyer?
94.2 FACTOID Who was the prosecutor?
94.3 FACTOID How did Mrs. McDougal plead?

94.4 LIST Who testified for Mrs. McDougal's defense?
94.5 FACTOID What was the jury's ruling on the obstruction of justice charge?
94.6 FACTOID What was the result of the contempt charges?
94.7 OTHER Other

Target 95 : return of Hong Kong to Chinese sovereignty
95.1 FACTOID What is Hong Kong's population?
95.2 FACTOID When was Hong Kong returned to Chinese sovereignty?
95.3 FACTOID Who was the Chinese President at the time of the return?
95.4 FACTOID Who was the British Foreign Secretary at the time?
95.5 LIST What other countries formally congratulated China on the return?
95.6 OTHER Other

Target 96 : 1998 Nagano Olympic Games
96.1 FACTOID What materials was the 1998 Olympic torch made of?
96.2 FACTOID How long was the men's downhill ski run in Nagano?
96.3 LIST Who won gold medals in Nagano?
96.4 FACTOID Which country took the first gold medal at Nagano?
96.5 FACTOID Who won the women's giant slalom?
96.6 FACTOID How many countries were represented at Nagano?
96.7 OTHER Other

Target 97 : Counting Crows
97.1 FACTOID Who is the lead singer of the Counting Crows?
97.2 FACTOID What year did the group form?
97.3 FACTOID What is the title of their signature hit?
97.4 FACTOID What is the title of the Crows' first record?
97.5 LIST List the Crows' record titles.
97.6 LIST List the Crows' band members.
97.7 OTHER Other

Target 98 : American Legion
98.1 FACTOID When was the American Legion founded?
98.2 FACTOID Where was the American Legion founded?
98.3 FACTOID How many members does the American Legion have?
98.4 FACTOID What organization has helped to revitalize Legion membership?
98.5 LIST List Legionnaires.
98.6 OTHER Other

Target 99 : Woody Guthrie
99.1 LIST List Woody Guthrie's songs.
99.2 FACTOID When was Guthrie born?
99.3 FACTOID Where was Guthrie born?
99.4 FACTOID What year did Woody Guthrie die?
99.5 FACTOID Where did he die?

99.6 FACTOID What caused Guthrie's death?
99.7 OTHER Other

Target 100 : Sammy Sosa
100.1 FACTOID Where was Sammy Sosa born?
100.2 FACTOID What was Sosa's team?
100.3 FACTOID How many home runs were hit by Sosa in 1998?
100.4 FACTOID Who was Sosa's competitor for the home run title in 1998?
100.5 FACTOID What was the record number of home runs in 1998?
100.6 FACTOID What award was won by Sammy Sosa in 1998?
100.7 LIST Name the pitchers off of which Sosa homered.
100.8 OTHER Other

Target 101 : Michael Weiss
101.1 FACTOID When was Michael Weiss born?
101.2 FACTOID Who is Weiss's coach?
101.3 FACTOID When did Weiss win his first U.S. Skating title?
101.4 FACTOID When did Weiss win his second U.S. Skating title?
101.5 FACTOID Who is Michael Weiss's choreographer?
101.6 FACTOID What is Weiss's home town?
101.7 LIST List Michael Weiss's competitors.
101.8 OTHER Other

Target 102 : Boston Big Dig
102.1 FACTOID What was the official name of the Big Dig?
102.2 FACTOID When did the Big Dig begin?
102.3 FACTOID What was the original estimated cost of the Big Dig?
102.4 FACTOID What was the expected completion date?
102.5 FACTOID What is the length of the Big Dig?
102.6 LIST List individuals associated with the Big Dig.
102.7 OTHER Other

Target 103 : Super Bowl XXXIV
103.1 FACTOID Where was Super Bowl XXXIV held?
103.2 FACTOID What team won the game?
103.3 FACTOID What was the final score?
103.4 FACTOID What was the attendance at the game?
103.5 FACTOID How many plays were there in Super Bowl XXXIV?
103.6 LIST List players who scored touchdowns in the game.
103.7 OTHER Other

Target 104 : 1999 North American International Auto Show
104.1 FACTOID In what city was the 1999 North American International Auto Show held?
104.2 FACTOID What type of vehicle dominated the show?
104.3 FACTOID What auto won the North American Car of the Year award at the show?

104.4 LIST List auto manufacturers in the show.
104.5 FACTOID How many automakers and suppliers had displays at the show?
104.6 FACTOID What was the expected attendance at the show?
104.7 FACTOID In what year was the first Auto Show held?
104.8 OTHER Other

Target 105 : 1980 Mount St. Helens eruption
105.1 FACTOID In what mountain range is Mt. St. Helens located?
105.2 FACTOID Who named Mount St. Helens?
105.3 FACTOID What was the date of Mt. St. Helens' eruption?
105.4 FACTOID How many people died when it erupted?
105.5 FACTOID What was the height of the volcano after the eruption?
105.6 LIST List names of eyewitnesses of the eruption.
105.7 OTHER Other

Target 106 : 1998 Baseball World Series
106.1 FACTOID What is the name of the winning team?
106.2 FACTOID What is the name of the losing team?
106.3 FACTOID Who was named Most Valuable Player (MVP)?
106.4 FACTOID How many games were played in the series?
106.5 FACTOID What is the name of the winning manager?
106.6 LIST Name the players in the series.
106.7 OTHER Other

Target 107 : Chunnel
107.1 FACTOID How long is the Chunnel?
107.2 FACTOID What year did construction of the tunnel begin?
107.3 FACTOID What year did the Chunnel open for traffic?
107.4 FACTOID How many people use the Chunnel each year?
107.5 FACTOID Who operates the Chunnel?
107.6 LIST List dates of Chunnel closures.
107.7 OTHER Other

Target 108 : Sony Pictures Entertainment (SPE)
108.1 FACTOID Who is the parent company of Sony Pictures?
108.2 FACTOID What U.S. company did Sony purchase to form SPE?
108.3 FACTOID Name the president and COO of the SPE.
108.4 LIST Name movies released by SPE.
108.5 LIST Name TV shows by the SPE.
108.6 FACTOID Who is the vice-president of SPE?
108.7 OTHER Other

Target 109 : Telefonica of Spain
109.1 FACTOID How many customers does it have?
109.2 FACTOID How many countries does it operate in?

109.3 FACTOID How is it ranked in size among the world's telecommunications companies?
109.4 FACTOID Name the chairman.
109.5 LIST Name companies involved in mergers with Telefonica of Spain.
109.6 OTHER Other

Target 110 : Lions Club International
110.1 FACTOID What is the mission of the Lions Club?
110.2 FACTOID When was the club founded?
110.3 FACTOID Where is the club's world-wide headquarters?
110.4 FACTOID Who is the Lions Club president?
110.5 LIST Name officials of the club.
110.6 LIST Name programs sponsored by the Lions Club.
110.7 OTHER Other

Target 111 : AMWAY
111.1 FACTOID When was AMWAY founded?
111.2 FACTOID Where is it headquartered?
111.3 FACTOID Who is the president of the company?
111.4 LIST Name the officials of the company.
111.5 FACTOID What is the name "AMWAY" short for?
111.6 OTHER Other

Target 112 : McDonald's Corporation
112.1 FACTOID When did the first McDonald's restaurant open in the U.S.?
112.2 FACTOID Where is the headquarters located?
112.3 FACTOID What is the corporation's annual revenue?
112.4 FACTOID Who made McDonald's the largest fast-food chain?
112.5 LIST Name the corporation's top officials.
112.6 LIST Name the non-hamburger restaurant holdings of the corporation.
112.7 OTHER Other

Target 113 : Paul Newman
113.1 FACTOID What is his primary career?
113.2 FACTOID What is his second successful career?
113.3 FACTOID What is the name of the company that he started?
113.4 LIST Name the camps started under his Hole in the Wall Foundation.
113.5 LIST Name some of his movies.
113.6 FACTOID Who is he married to?
113.7 OTHER Other

Target 114 : Jesse Ventura
114.1 FACTOID What is his political party affiliation?
114.2 FACTOID What is his birth name?
114.3 LIST List his various occupations.

114.4 LIST Name movies/TV shows he appeared in.
114.5 FACTOID What is his wife's name?
114.6 FACTOID How many children do they have?
114.7 OTHER Other

Target 115 : Longwood Gardens
115.1 FACTOID When was the initial land purchased?
115.2 FACTOID Where is it?
115.3 FACTOID How large is it?
115.4 FACTOID Who created it?
115.5 FACTOID How many visitors does it get per year?
115.6 FACTOID When is the best month to visit the gardens?
115.7 LIST List personnel of the gardens.
115.8 OTHER Other

Target 116 : Camp David
116.1 FACTOID Where is it?
116.2 FACTOID How large is it?
116.3 FACTOID What was it originally called?
116.4 FACTOID When was it first used?
116.5 FACTOID What U.S. President first used it?
116.6 LIST Who are some world leaders that have met there?
116.7 OTHER Other

Target 117 : kudzu
117.1 FACTOID What kind of plant is kudzu?
117.2 FACTOID When was it introduced into the U.S.?
117.3 FACTOID From where was it introduced?
117.4 LIST What are other names it is known by?
117.5 FACTOID Why is it a problem?
117.6 FACTOID What has been found to kill it?
117.7 OTHER Other

Target 118 : U.S. Medal of Honor
118.1 FACTOID When was it first awarded?
118.2 FACTOID Who authorized it?
118.3 FACTOID How many have received the award since 1863?
118.4 LIST What Medal of Honor recipients are in Congress?
118.5 FACTOID Who is the only woman to receive it?
118.6 FACTOID How many veterans have received the honor twice?
118.7 OTHER Other

Target 119 : Harley-Davidson
119.1 FACTOID When was the company founded?
119.2 FACTOID Where is it based?

119.3 FACTOID They are best known for making what product?
119.4 LIST What other products do they produce?
119.5 FACTOID What is the average age of a Harley-Davidson rider?
119.6 FACTOID What company did Harley-Davidson buy out?
119.7 OTHER Other

Target 120 : Rose Crumb
120.1 FACTOID What was her occupation?
120.2 FACTOID Where was she from?
120.3 FACTOID What organization did she found?
120.4 FACTOID When did she found it?
120.5 LIST What awards has she received?
120.6 FACTOID How old was she when she won the awards?
120.7 OTHER Other

Target 121 : Rachel Carson
121.1 FACTOID What was her vocation?
121.2 FACTOID Where was her home?
121.3 LIST What books did she write?
121.4 FACTOID When did she write her book exposing dangers of pesticides?
121.5 FACTOID Her book caused what pesticide to be banned?
121.6 FACTOID What did she die of?
121.7 FACTOID When did she die?
121.8 OTHER Other

Target 122 : Paul Revere
122.1 FACTOID When was he born?
122.2 FACTOID When did he die?
122.3 FACTOID In what cemetery is he buried?
122.4 FACTOID When did he make his famous ride?
122.5 FACTOID From where did he begin his famous ride?
122.6 FACTOID Where did his famous ride end?
122.7 LIST What were some of his occupations?
122.8 OTHER Other

Target 123 : Vicente Fox
123.1 FACTOID When was Vicente Fox born?
123.2 FACTOID Where was Vicente Fox educated?
123.3 FACTOID Of what country is Vicente Fox president?
123.4 FACTOID What job did he hold before becoming president?
123.5 LIST What countries did Vicente Fox visit after election?
123.6 OTHER Other

Target 124 : Rocky Marciano
124.1 FACTOID When was he born?

124.2 FACTOID Where did he live?
124.3 FACTOID When did he die?
124.4 FACTOID How did he die?
124.5 FACTOID How many fights did he win?
124.6 LIST Who were some of his opponents?
124.7 OTHER Other

Target 125 : Enrico Caruso
125.1 LIST What operas has Caruso sung?
125.2 FACTOID Whom did he marry?
125.3 FACTOID How many children did he have?
125.4 FACTOID How many opening season performances did he have at the Met?
125.5 FACTOID How many performances did he sing at the Met?
125.6 FACTOID At what age did he die?
125.7 FACTOID Where did he die?
125.8 OTHER Other

Target 126 : Pope Pius XII
126.1 FACTOID When was he elected Pope?
126.2 FACTOID What was his name before becoming Pope?
126.3 LIST What official positions did he hold prior to becoming Pius XII?
126.4 FACTOID How long was his pontificate?
126.5 FACTOID How many people did he canonize?
126.6 FACTOID What caused the death of Pius XII?
126.7 FACTOID What pope followed Pius XII?
126.8 OTHER Other

Target 127 : U.S. Naval Academy
127.1 FACTOID Where is the U.S. Naval Academy?
127.2 FACTOID When was it founded?
127.3 FACTOID What is the enrollment?
127.4 FACTOID What are the students called?
127.5 FACTOID Who is the father of the U.S. Navy?
127.6 LIST List people who have attended the Academy.
127.7 OTHER Other

Target 128 : OPEC
128.1 FACTOID What does OPEC stand for?
128.2 FACTOID How many countries are members of OPEC?
128.3 LIST What countries constitute the OPEC committee?
128.4 FACTOID Where is the headquarters of OPEC located?
128.5 LIST List OPEC countries.
128.6 OTHER Other

Target 129 : NATO

129.1 FACTOID What does the acronym NATO stand for?
129.2 FACTOID When was NATO established?
129.3 FACTOID Where was the agreement establishing NATO signed?
129.4 LIST Which countries were the original signers?
129.5 FACTOID Where is NATO headquartered?
129.6 OTHER Other

Target 130 : tsunami
130.1 FACTOID What causes tsunamis?
130.2 FACTOID Where does it commonly occur?
130.3 FACTOID What is its maximum height?
130.4 FACTOID How fast can it travel?
130.5 LIST What countries has it struck?
130.6 FACTOID What language does the term "tsunami" come from?
130.7 OTHER Other

Target 131 : Hindenburg disaster
131.1 FACTOID What type of craft was the Hindenburg?
131.2 FACTOID How fast could it travel?
131.3 FACTOID When did the Hindenburg disaster occur?
131.4 FACTOID Where did the disaster occur?
131.5 FACTOID How many people were on board?
131.6 FACTOID How many of them were killed?
131.7 LIST Name individuals who witnessed the disaster.
131.8 OTHER Other

Target 132 : Kim Jong Il
132.1 FACTOID When was Kim Jong Il born?
132.2 FACTOID Who is Kim Jong Il's father?
132.3 FACTOID What country does Kim Jong Il rule?
132.4 LIST What posts has Kim Jong Il held in the government of this country?
132.5 FACTOID To whom is Kim Jong Il married?
132.6 OTHER Other

Target 133 : Hurricane Mitch
133.1 FACTOID Where did this hurricane occur?
133.2 FACTOID When did this hurricane occur?
133.3 LIST As of the time of Hurricane Mitch, what previous hurricanes had higher death totals?
133.4 LIST What countries offered aid for this hurricane?
133.5 FACTOID What country had the highest death total from this hurricane?
133.6 OTHER Other

Target 134 : genome
134.1 FACTOID What is a genome?

134.2 LIST List species whose genomes have been sequenced.
134.3 LIST List the organizations that sequenced the Human genome.
134.4 FACTOID How many chromosomes does the Human genome contain?
134.5 FACTOID What is the length of the Human genome?
134.6 OTHER Other

Target 135 : Food-for-Oil Agreement
135.1 FACTOID What country was the primary beneficiary of this agreement?
135.2 FACTOID Who authorized this agreement?
135.3 FACTOID When was this agreement authorized?
135.4 FACTOID When was this agreement signed?
135.5 LIST What countries participated in this agreement by providing food or medicine?
135.6 OTHER Other

Target 136 : Shiite
136.1 FACTOID Who was the first Imam of the Shiite sect of Islam?
136.2 FACTOID Where is his tomb?
136.3 FACTOID What was this person's relationship to the Prophet Mohammad?
136.4 FACTOID Who was the third Imam of Shiite Muslims?
136.5 FACTOID When did he die?
136.6 FACTOID What portion of Muslims are Shiite?
136.7 LIST What Shiite leaders were killed in Pakistan?
136.8 OTHER Other

Target 137 : Kinmen Island
137.1 FACTOID What is the former name of Kinmen?
137.2 FACTOID What country governs Kinmen?
137.3 LIST What other island groups are controlled by this government?
137.4 FACTOID In the 1950's, who regularly bombarded Kinmen?
137.5 FACTOID How far is Kinmen from this country?
137.6 FACTOID Of the two governments involved over Kinmen, which has air superiority?
137.7 OTHER Other

Target 138 : International Bureau of Universal Postal Union (UPU)
138.1 FACTOID When was the UPU organized?
138.2 FACTOID When did the UPU become part of the UN?
138.3 LIST Where were UPU congresses held?
138.4 FACTOID When did China first join the UPU?
138.5 FACTOID Who is the Director-General of the UPU?
138.6 OTHER Other

Target 139 : Organization of Islamic Conference (OIC)
139.1 FACTOID When was the Organization of Islamic Conference organized?
139.2 LIST Which countries are members of the OIC?

139.3 LIST Who has served as Secretary General of the OIC?
139.4 FACTOID Where was the 8th summit of the OIC held?
139.5 FACTOID Where was the 24th Islamic Conference of Foreign Ministers of the OIC held?
139.6 OTHER Other

Target 140 : PBGC
140.1 FACTOID What government organization goes by the acronym PBGC?
140.2 FACTOID Who is the head of PBGC?
140.3 FACTOID When was PBGC established?
140.4 LIST Employees of what companies are receiving benefits from this organization?
140.5 FACTOID What is the average waiting time for this organization to determine benefits?
140.6 OTHER Other

APPENDIX C: TREC 2005 QA FACTOID ANSWERS

66.1 (Aug(.|ust)? 12,?)?2000
66.2 (Capt(.?|ain))?(Gennady)?Lyachin
66.3 Northern(Fleet)?
66.4 118(crewmen)?
66.6 Barents(Sea)?
67.1 (Lara)?Dutta
67.2 India
67.3 78
67.4 (Nicosia|Cyprus)
67.5 May.*13(2000)?
68.1 (Tasmania|Australia)
68.2 (April 28,?)?1996
68.3 (32|35)((persons|people))?
68.4 (Martin)?Bryant
68.5 Australian?
69.1 (July 12,?)?1998
69.2 Brazil
69.3 (3 ?- ?0|0-3)
69.4 les Bleus
69.5 (Stade de France|Stadium of France)
69.6 Aime Jacquet
70.1 (Feb.? 3,?)?1998
70.2 (Cavalese|Mount Cermis|Italian Alps)
70.3 (20|twenty)(people)?
70.4 (U.S.|U . S .|US|Marine Corps|United States)
70.5 (Richard)?Ashby
70.6 acquitt(ed|al)
71.1 fighter((bombers?|planes?))?
71.2 400 to 500 miles per hour
71.3 Lockheed(-Martin)?
71.4 Bethesda ?,? (Maryland|Md)
71.5 (Pratt (and|&) Whitney|General Electric)
72.1 (Bombay|Mumbai|India)
72.2 (Bombay|Mumbai)
72.3 Malabar Hill
72.4 International India Film Awards
72.5 (2nd|second)
73.1 (impotence?|(erectile|sexual).*dysfunctions?)
73.2 Pfizer(Pharmaceuticals)?(Inc.?)?
73.3 State Drug Administration(of China)?
73.4 (S|s)ildenafil citrate
73.5 (April)?1998
74.1 ([cC]ollege|[uU]niversity|liberal arts)
74.2 (Greencastle(,? (Ind.?|Indiana))?|Ind.?|Indiana)
74.3 1837

74.4 (Robert)?(G.?)?Bottoms
74.5 (2,?200|2,334)
75.1 (Whitehouse Station(,? (N.J.|New Jersey))?|N.?J.?|New Jersey)
75.2 (drugs?|Vioxx|pharmaceutics?)
75.3 MRK
75.4 www.merck.com
75.6 Raymond V Gilmarman
76.1 (singer|actor|entertainer)
76.2 Going My Way
76.3 Der Bingle
76.4 White Christmas
76.5 Gonzaga
76.6 (74|73)
77.1 (Jan.? 10,? |January 10,?)?1949
77.2 (Marshall|Texas)
77.3 1973
77.4 Joe Frazier
78.1 (Sep.? 6,? |September 6,?)?1998
78.2 (March 23,?)?1910
78.4 ((movie |film)?director|filmmaker)
78.5 (The)?Emperor
78.6 actress
79.1 (May 21,?)?1998
79.2 (22|two ?dozen|25|Twenty-five)
79.4 (two|2)(students)?
79.5 15(-year)?
79.6 (five|5)(bombs)?
80.1 Nantucket(Island)?(,? Mass(.?|achusetts))?
80.2 (Ahmed(Mahmoud)?)?(el-|El |al)Habash(y|i)
80.3 ((Gamil|Gameel)(Hamid)?)?(Al-|al-|el-|El-|El)?Batout(y|i)
80.4 (15|17|18)(crew members)?
80.5 (197|199)(passengers)?
81.1 Real Quiet
81.3 (Bob)?Baffert
81.4 Victory Gallop
81.5 103,269
82.1 1947
82.2 (Sept.?|September) 30,? 1960
82.5 Elmer
83.1 palace
83.2 1793
83.5 (3|3.1|3.2|3.6|5.1|6) million
83.6 Pierre Rosenberg
84.1 Willamette?(Meteorite)?
84.2 (14 metric tons|15-ton)

84.3 Tomanoas
84.4 Namibia
84.5 60(-|)ton(s)?
84.6 910(metric tons)?
85.2 (Carnival|Carnival(Cruise Lines|Corp.?)|Star(Cruises)?)
85.3 Great Stirrup Cay
85.4 fourth(-|)largest
85.5 chartered for gay passengers
86.1 (June 1993|November 1993)
86.2 heart attack
86.3 54
86.4 (Abdulsalam)?Abubakar
87.1 (Sept(.|ember) 29,?)?1901
87.2 (Nov(.|ember) 28,?)?1954
87.3 Physics
87.5 (first controlled nuclear chain reaction|first atom bomb)
87.6 fermium,? Fm
87.7 Italy
88.1 Atlanta
88.2 (James P.|Jim) Kelly
88.3 (Nov(.?|ember) 10,?)?1999
88.5 brown
88.6 (\$ 99.8 million|about $22 million)
89.1 (Williamsport|Pa|Pennsylvania)
89.2 Windsor Ave.?
89.4 (fifth|five|5)
89.5 1947
89.6 (http://)?www.littleleague.org
90.2 1,500(acres)?
90.4 (Thomas)?Jefferson
90.6 Antony Champ
91.1 Cliff Hillegass
91.2 (William)?Shakespeare
91.4 IDG(Books(Worldwide)?)?
91.5 5 million(copies)?
92.1 (four|4)(times)?
92.2 (two|2)
92.5 PGA(Championship)?
92.6 (Winnie|Winifred)
93.1 Jim Lehrer
93.2 90(|\-)minutes?
93.3 (University of)?Massachusetts
93.4 NBC
93.5 Florida
93.6 Oct(.|ober) 3

94.1 Mark Geragos
94.2 (Mark Barrett|Julie Myers|Kenneth Starr|W. Hickman Ewing Jr.)
94.3 innocent
94.5 (acquitt(ed|al)|innocent)
94.6 deadlock(ed)?
95.1 ((six|6)(.(4|8))? million|6,782,100)
95.2 (July 1,?)?1997
95.3 Jiang(Zemin)
95.4 Robin Cook
96.1 pine and bamboo
96.2 (2,?923 meters|2,920-meter)
96.4 Russia
96.5 ((Deborah)?Compagnoni|Karine Ruby)
96.6 (70|71|72|82)
97.1 Adam Duritz
97.2 1990
97.3 Mr. Jones
97.4 August and Everything After
98.1 (March 15,?)?1919
98.2 Paris
98.3 ((2.8|2.9) million|2800000)
98.4 Sons of the Legion
99.2 (July 14,?)?1912
99.3 (Okla(homa)?|Okemah(,? (Okla.?|Oklahoma))?)
99.4 1967
99.5 New York
99.6 Huntington's((chorea|disease))?
100.1 (Santo Domingo|Dominican(Republic)?)
100.2 (Chicago)?Cubs
100.3 66
100.4 (Mark)?McGwire
100.5 70
100.6 ((National League|NL))?(Most Valuable Player|MVP|most-valuable-player)(award)?
101.1 (1976|1977)
101.2 Audrey Weisiger
101.3 1999
101.4 2000
101.5 Lisa (Thornton|Weiss)
101.6 Fairfax(,? (VA|Virginia))?
102.1 Central Artery ?(/|-)? ?(Third Harbor)?(Tunnel)?Project
102.2 1991
102.3 (\$ 10.8 billion|10.8 billion dollars?|\$?2.6 billion)
102.4 (2004|(early)?2005)
102.5 (three|3|71/2|7.5)(|-)miles?
103.1 Atlanta

103.2 (St. Louis)?Rams
103.3 (23 ?-? ?16|16-23)
103.4 (72,?000|72,652)
103.5 132(plays)?
104.1 Detroit
104.2 trucks
104.3 Beetle
104.5 48
104.6 800,000
104.7 1907
105.1 Cascade((M|m)ountains)?
105.2 (George)?Vancouver
105.3 May 18 ?,? 1980
105.4 57(people)?
105.5 8,363 feet
106.1 (New York)?Yankees
106.2 (San Diego(Padres)?|Padres)
106.3 (Scott)?Brosius
106.4 (four|4)
106.5 (Joe)?Torre
107.1 (31|32) miles
107.2 1987
107.3 1994
107.4 millions
107.5 Eurotunnel
108.1 Sony(Corporation)?
108.2 (Columbia|Tristar)(Pictures)?
108.3 ((John)?Calley|(Bob)?Wynne|(Mel)?Harris|Sagansky)
108.6 (Patrick Kennedy|Masayuki Nozoe)
109.1 (55|62) million(customers)?
109.2 17
109.3 12th(largest)?
109.4 ((Juan)?Villalonga|Alierta)
110.1 (aid|help)(ing)? (for|the)? blind(and visually impaired)?(people)?
110.2 (1917|1937)
110.4 ((Kajit)?Habanananda|(James)?Ervin|Augustin Soliva)
111.1 1959
111.2 (Ada)?(,? ?Mich(.?|igan))
111.3 ((Richard|Dick))?DeVos
112.1 (1940|1955)
112.2 (Oak Brook(,? Ill(.?|inois))?|Illinois)
112.3 \$?(31.8|33|35|36) billion
112.4 (Ray)?Kroc
113.1 actor
113.2 (racing|race-car driver)

113.3 Newman ?'?s ?Own
113.6 (Joanne)?Woodward
114.1 Reform(Party)?
114.2 (James(George)?)?Janos
114.5 Terry(Ventura)?
114.6 (two|2)(children)?
115.1 1906
115.2 Kennett Square(,? (Pa.?|Pennsylvania))?
115.3 (thousand|1,050)(?- ?|)acres?
115.4 ((Pierre)((S.?|Samuel)))?(d|D)u Pont
115.5 (a million|1,000,000|800,000)(visitors)?
115.6 May
116.1 ((Thurmont|Catoctin Mountains),?)?(Maryland|Md.?)
116.2 143(-|)acres?
116.3 Shangri-La
116.4 (during World War II|1942|1943)
116.5 (Franklin(D.?)?)?Roosevelt
117.1 (weed|vine)
117.2 (1800s|1920s|1876)
117.3 (Asia|Chin(a|ese)|Japan)
117.5 (smother(ing|s)|grows over) everything
117.6 (Myrothecium verrucaria|fungus|sicklepod|herbicides)
118.1 1862
118.2 Congress
118.3 (3,399|3,408|3,410)(people)?
118.5 (Dr.?)?(Mary)?(Edwards)?Walker
118.6 nineteen
119.1 1903
119.2 Milwaukee(,? Wisconsin)?
119.3 motorcycles?
119.5 40 years
119.6 AMF
120.1 (nurse|hospice care crusader)
120.2 Port Angeles(,? Wash(.?|ington))?
120.3 Hospice of Clallam County
120.4 1978
120.6 72
121.1 ((marine)?biologist|author)
121.2 Springdale(,? (Pa|Pennsylvania))?
121.4 1962
121.5 DDT
121.6 (breast)?cancer
121.7 1964
122.1 ((January 1,?)?1735|(January 7,?)?1942)
122.2 1818

122.3 Granary (Bur(ying|ial))?Ground
122.4 (April 18,?)?1775
122.5 (Boston|Charlestown(,? Mass(.?|achusetts))?)
122.6 Lexington(,? Mass(.?|achusetts))?
123.1 (July 2,?)?1942
123.2 (Universidad Iberoamericana|Harvard(University)?)
123.3 Mexico
123.4 governor(of Guanajuato(state)?)?
124.1 (Sept(.?|ember) 1,?)?1923
124.2 Brockton(,? Mass(.?|achusetts))?
124.3 (Aug(.?|ust) 31,?)?1969
124.4 (plane)?crash
124.5 49(fights)?
125.2 (Dorothy(Park)?)?Benjamin
125.3 (four|five|4|5)(children)?
125.4 17
125.5 607 performances
125.6 48
126.1 (March 2,?)?1939
126.2 (Eugenio)?Pacelli
126.4 19((-|)years?(,? seven months and seven days))?
126.6 stroke
126.7 John XXIII
127.1 Annapolis(,? (Md.?|Maryland))?
127.2 1845
127.3 4,000
127.4 midshipm(a|e)n
127.5 ((Commodore)?(John)?Barry|(John Paul)?Jones)
128.1 (Oil Producing and|Orga?nization ?(of ?(the)?)?(Oil|Petroleum)) Exporting Countries
128.2 (11|13)
128.4 (Vienna|Austria)
129.1 North Atlantic Treaty Organi(s|z)ation
129.2 (April 4,?)?1949
129.3 Washington
129.5 (Brussels(,? Belgium)?|Belgium)
130.1 ((undersea|underwater)? ?earthquakes?|landslides|volcanoe?s)
130.2 Pacific Ocean
130.3 (100 feet \(30 meters\)|100 feet|30 meters)
130.4 (450|500) (mph|miles per hour|milesperhour)
130.6 Japanese
131.1 (Zeppelin|airship|dirigible|(Zeppelin)?(dirigible)?(passenger)?airship)
131.2 (80 miles|90 mph)
131.3 (May 6(th)?,?)?1937
131.4 (Lakehurst,? ?)?(N.J.|New Jersey)?
131.5 9[67](people)?

131.6 (3[56](people)?|35 people on board and one on the ground|35 of the 97 people on board and a Navy crewman on the ground)
132.1 (February 16,?)?1942
132.2 Kim Il S(o|u)ng
132.3 (North Korea|Democratic People's Republic of Korea|DPRK)
132.5 Kim Young Sook
133.1 (Atlantic|Caribbean|Central America|(Honduras and Nicaragua)|Honduras|Nicaragua|Guatemala|El Salvador)
133.2 (((the last week of |late)?October,?)?1998|late October and early November)
133.5 Honduras
134.1 (DNA(sequence)?|genetic blueprint|a complete haploid set of chromosomes of an organism)
134.4 (23(pairs of chromosomes)?|46(chromosomes)?)
134.5 ((3|four) billion(units of DNA| DNA units| chemical units)?|(80|100),000 genes)
135.1 (Iraq|China)
135.2 (U.?N.?|United Nations)
135.3 (April 1995|(December,?)?1996)
135.4 (May,? (20)?)?1996
136.1 (Talib|((Imam)?Ali((Ibn|Bin) (Abi|Abu) Talib(\(AS\))?)?))
136.2 (Iraqi?|Najaf(,? Iraq)?)
136.3 son[-]in[-]law
136.4 (Imam)?Hussein
136.5 (1,000 years ago|680)
136.6 (10 percent|about 15%)
137.1 Quemoy
137.2 Taiwan
137.4 China
137.5 (a mile|(((two|2|1.8) miles)|((3|three) (km|kilometers)(\(1.8 miles\))?)))
137.6 (China|Taiwan)
138.1 1874
138.2 1948
138.4 (March 1,?)?1914
138.5 Thomas (E.)?Leavey
139.1 (1969|(March|May) 1970)
139.4 Tehran(,? (capital of)?Iran)?
139.5 (Jakarta(,? Indonesia)?|Indonesia)
140.1 Pension Benefits? Guaranty Corp(.|oration)?
140.2 David (M.)?Strauss
140.3 1974
140.5 3.2 years

REFERENCES

Abductive Inference Page. 1997. Abductive Inference in Reasoning and Perception. http://www.cis.ohio-state.edu/lair/Projects/Abduction/abduction.html.

Abney, S. 1996. Partial Parsing via Finite-State Cascades. In Proceedings of Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information, 8-15. Prague, Czech Republic.

Ahn, K.; Bos, J.; Clark, S.; Curran, J.; Dalmas, T.; Leidner, J.; Smillie, M.; and Webber, B. 2005. Question Answering with QED and Wee at TREC-2004. In Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), NIST Special Publication 500-261. Gaithersburg, MD: National Institute of Standards and Technology.

Appelt, D.E. and Israel, D.J. 1999. Introduction to Information Extraction Technology. A Tutorial Prepared for the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 1999.

Brill, E. 1994. Some Advances in Part of Speech Tagging. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94). Seattle, Washington: American Association for Artificial Intelligence.

Fellbaum, C. ed. 1998. WordNet: An Electronic Lexical Database. Cambridge, Mass.: The MIT Press.

Gaizauskas, R.; Greenwood, M.; Hepple, M.; Roberts, I.; and Saggion, H. 2005. The University of Sheffield's TREC 2004 Q&A Experiments. In Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), NIST Special Publication 500-261. Gaithersburg, MD: National Institute of Standards and Technology.

Glinos, D. 1999. An Intelligent Editor for Natural Language Processing of Unrestricted Text, Master's Thesis, Department of Computer Science, University of Central Florida, Orlando, Florida, 1999.

Gomez, F. 2000. Why Base the Knowledge Representation Language on Natural Language? In Journal of Intelligent Systems, Vol.10, No.2, 2000, Freund and Pettman Publishers.

Gomez, F. 2001. An Algorithm for Aspects of Semantic Interpretation Using an Enhanced WordNet. In Proceedings of the North American Chapter of the Association for Computational Linguistics annual meeting (NAACL 2001), June 2001, Pittsburgh, Pennsylvania.

Gomez, F. 2004a. Building Verb Predicates: A Computational View. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL-04) pp. 359-366. Barcelona, Spain, July 2004.

Gomez, F. 2004b. Grounding the Ontology on the Semantic Interpretation Algorithm. In Proceedings of the Second Global WordNet Conference (GWC) pp. 124-129. Masaryk University, Brno, Czech Republic, January 2004.

Greenwood, M.; Roberts, I.; and Gaizauskas, R. 2003. The University of Sheffield TREC 2002 Q&A System. In Proceedings of the Eleventh Text REtrieval Conference (TREC 2002), NIST Special Publication 500-251. Gaithersburg, MD: National Institute of Standards and Technology.

Greenwood, M., Saggion, H. 2004. A Pattern Based Approach to answering Factoid, List and Definition Questions. In: Proceeding of the 7th RIAO Conference (RIAO 2004), Avignon, France, April 27, 2004.

Harabagiu, S.; Moldovan, D.; Clark, C.; Bowden, M.; Williams, J.; and Bensley, J. 2004. Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In Proceedings of the Twelfth Text REtrieval Conference (TREC 2003), 375-382. NIST Special Publication 500-255. Gaithersburg, MD: National Institute of Standards and Technology.

Hildebrandt, W., Katz, B., and Lin, J. 2004. Answering Definition Questions Using Multiple Knowledge Sources. In: Proceedings of the 2004 Human Language Technology conference / North American Chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL 2004), May 2-7, 2004, Boston, Massachusetts.

Kies, D. 2004. Modern English Grammar, In *The HyperTextBooks*, Department of English, College of DuPage, Glen Ellyn, Illinois http://www.papyr.com/hypertextbooks/engl_126

Litkowski, K. 2003. Question Answering Using XML-Tagged Documents. In Proceedings of the Eleventh Text REtrieval Conference (TREC 2002). NIST Special Publication 500-251. Gaithersburg, MD: National Institute of Standards and Technology.

Miller, G.A. 1995. WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, No. 11, pp. 39-41, November 1995.

Montes-y-Gomez, M.; Lopez-Lopez, A.; and Gelbukh, A. 2000. Information Retrieval with Conceptual Graph Matching. In Proceedings of the 11th International Conference and Workshop on Database and Expert Systems Applications (DEXA-2000), Lecture Notes in Computer Science, N 1873, Springer-Verlag, pp. 312-321. Greenwich, England, September 2000.

Riloff, E. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In Proceeding of Eleventh National Conference on Artificial Intelligence, 1993, pages 811-816.

Riloff, E, and Jones, R. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In Proceeding of the Sixtenth National Conference on Artificial Intelligence (AAAI-99), 474-479.

Schone, P.; Ciany, G.; McNamee, P.; Mayfield, J.; Bassi, T.; and Kulman, A. 2005. Question Answering with QACTIS at TREC-2004. In Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), NIST Special Publication 500-261. Gaithersburg, MD: National Institute of Standards and Technology.

Voorhees, E.M. 2005a: Overview of TREC 2004. In Voorhees, E.M., and Buckland, L.P. eds.: Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), NIST Special Publication 500-261. Gaithersburg, MD: National Institute of Standards and Technology.

Voorhees, E.M. 2005b: Overview of the TREC 2004 Question Answering Track. In Voorhees, E.M., and Buckland, L.P. eds.: Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004), NIST Special Publication 500-261. Gaithersburg, MD: National Institute of Standards and Technology.

Yang, H., Cui, H., Kan, M., Maslennikov, M., Qui, L., and Chua, T. 2004. In Proceedings of the Twelfth Text REtrieval Conference (TREC 2003). NIST Special Publication 500-255. Gaithersburg, MD: National Institute of Standards and Technology.